

53-1001341-02
August 7, 2009



Fabric OS Encryption

Administrator's Guide

Supporting Fabric OS v6.3.0

BROCADE

Copyright © 2008-2009 Brocade Communications Systems, Inc. All Rights Reserved.

Brocade, the B-wing symbol, BigIron, DCX, Fabric OS, FastIron, IronPoint, IronShield, IronView, IronWare, JetCore, NetIron, SecureIron, ServerIron, StorageX, and Turbolron are registered trademarks, and DCFM, Extraordinary Networks, and SAN Health are trademarks of Brocade Communications Systems, Inc., in the United States and/or in other countries. All other brands, products, or service names are or may be trademarks or service marks of, and are used to identify, products or services of their respective owners.

Notice: This document is for informational purposes only and does not set forth any warranty, expressed or implied, concerning any equipment, equipment feature, or service offered or to be offered by Brocade. Brocade reserves the right to make changes to this document at any time, without notice, and assumes no responsibility for its use. This informational document describes features that may not be currently available. Contact a Brocade sales office for information on feature and product availability. Export of technical data contained in this document may require an export license from the United States government.

The authors and Brocade Communications Systems, Inc. shall have no liability or responsibility to any person or entity with respect to any loss, cost, liability, or damages arising from the information contained in this book or the computer programs that accompany it.

The product described by this document may contain "open source" software covered by the GNU General Public License or other open source license agreements. To find out which open source software is included in Brocade products, view the licensing terms applicable to the open source software, and obtain a copy of the programming source code, please visit <http://www.brocade.com/support/oscd>.

Brocade Communications Systems, Incorporated

Corporate and Latin American Headquarters
Brocade Communications Systems, Inc.
1745 Technology Drive
San Jose, CA 95110
Tel: 1-408-333-8000
Fax: 1-408-333-8101
E-mail: info@brocade.com

Asia-Pacific Headquarters
Brocade Communications Systems China HK, Ltd.
No. 1 Guanghua Road
Chao Yang District
Units 2718 and 2818
Beijing 100020, China
Tel: +8610 6588 8888
Fax: +8610 6588 9999
E-mail: china-info@brocade.com

European Headquarters
Brocade Communications Switzerland Sàrl
Centre Swissair
Tour B - 4ème étage
29, Route de l'Aéroport
Case Postale 105
CH-1215 Genève 15
Switzerland
Tel: +41 22 799 5640
Fax: +41 22 799 5641
E-mail: emea-info@brocade.com

Asia-Pacific Headquarters
Brocade Communications Systems Co., Ltd. (Shenzhen WFOE)
Citic Plaza
No. 233 Tian He Road North
Unit 1308 - 13th Floor
Guangzhou, China
Tel: +8620 3891 2000
Fax: +8620 3891 2111
E-mail: china-info@brocade.com

Document History

Title	Publication number	Summary of changes	Date
<i>Fabric OS Encryption Administrator's Guide</i>	53-1001114-01	New document.	August 2008
<i>Fabric OS Encryption Administrator's Guide</i>	53-1001114-02	Revised document to include additional best practices.	September 2008
<i>Fabric OS Encryption Administrator's Guide</i>	53-1001114-03	Revised document to include new performance licensing information.	September 2008
<i>Fabric OS Encryption Administrator's Guide</i>	53-1001201-01	Revised document for Fabric OS version 6.2.0.	November 2008
<i>Fabric OS Encryption Administrator's Guide</i>	53-1001201-02	Revised document to synchronize with DCFM version 10.1.0.	December 2008
<i>Fabric OS Encryption Administrator's Guide</i>	53-1001201-03	Revised document to incorporate changes to key manager software procedures.	March 2009
<i>Fabric OS Encryption Administrator's Guide</i>	53-1001341-01	Revised document for Fabric OS version 6.3.0.	July 2009
<i>Fabric OS Encryption Administrator's Guide</i>	53-1001341-02	Revised document to incorporate support for Virtual Fabrics, KAC login information for HP and Thales key vaults, and other various updates.	August 2009

Contents

About This Document

In this chapter	xiii
How this document is organized	xiii
Supported hardware and software	xiv
What's new in this document	xiv
Document conventions	xiv
Notice to the reader	xvi
Additional information	xvi
Getting technical help	xvii
Document feedback	xviii

Chapter 1

Encryption overview

In this chapter	1
Host and LUN considerations	1
Encryption configuration tasks	2
Terminology	3
The Brocade encryption switch	5
The FS8-18 blade	6
Performance licensing	6
Adding a license	6
Licensing best practices	6
Recommendation for connectivity	7
Usage limitations	7
Brocade encryption solution overview	8
Data flow from server to storage	9
Data encryption key life cycle management	10
Key management systems	12
Master key management	12
Encryption switch initialization	13
Exporting, importing, and loading certificates	13
Support for Virtual Fabrics	13

Chapter 2

Encryption configuration using the Management application

In this chapter	15
Gathering information.	16
Encryption user privileges	17
Encryption Center features.	18
Smart card usage	18
Registering authentication cards from a card reader	19
Registering authentication cards from the database	20
De-registering an authentication card	20
Using authentication cards	20
Registering system cards from a card reader	21
De-registering a system card.	21
Enabling or disabling the system card requirement	21
Viewing and editing switch encryption properties	22
Saving the public key certificate	24
Enabling the encryption engine state.	24
Viewing and editing group properties	25
General tab.	26
Members tab	26
Consequences of removing an encryption switch.	27
Security tab	29
HA Clusters tab.	29
Engine Operations tab	30
Link Keys tab	31
Tape Pools tab	32
Tape pools overview.	32
Adding tape pools	33
Encryption Targets dialog box.	34
Redirection zones	36
Creating a new encryption group	37
Adding a switch to an encryption group.	47
Creating high availability (HA) clusters	50
Removing engines from an HA cluster	51
Swapping engines in an HA cluster	52
Failback option.	52
Invoking failback	53
Adding encryption targets.	54
Configuring hosts for encryption targets	61
Adding Target Disk LUNs for encryption	62
Adding Target Tape LUNs for encryption	65
Configuring encrypted storage in a multi-path environment	66

Master keys	67
Active master key	67
Alternate master key	68
Master key actions	68
Reasons master keys can be disabled	68
Saving the master key to a file	68
Saving a master key to a key vault	70
Saving a master key to a smart card set	71
Restoring a master key from a file	73
Restoring a master key from a key vault	74
Restoring a master key from a smart card set	75
Creating a new master key	76
Zeroizing an encryption engine	77
Tracking Smart Cards	79
Encryption-related acronyms in log messages	80

Chapter 3

Encryption configuration using the CLI

In this chapter	81
Overview	81
Command validation checks	82
Command RBAC permissions and AD types	83
Cryptocfg Help command output	86
Setting default zoning to no access	87
Management port configuration	87
I/O sync link configuration	88
Assigning static IP addresses to Ge0 and Ge1	88
Special consideration for blades	88
IP Address change of a node within an encryption group	89
Encryption switch initialization	90
Initializing an encryption switch	90
Checking encryption engine status	92
Certificate Exchange	93
Exporting a certificate	93
Importing a certificate	94
Viewing imported certificates	95
Basic encryption group configuration	95
Creating an encryption group	95
Setting the key vault type	96
Adding a member node to an encryption group	96
Group-wide policy configuration	98
Key vault configuration	99
High Availability (HA) cluster configuration	100
HA cluster configuration rules	100
Creating an HA cluster	101
Adding an encryption engine to an HA cluster	101

CryptoTarget container configuration	102
Gathering information	103
Frame redirection	103
Creating an initiator - target zone	104
Creating a CryptoTarget container	105
Removing an initiator from a CryptoTarget container	107
Deleting a CryptoTarget container	107
Moving a CryptoTarget container	108
Crypto LUN configuration	109
Discovering a LUN	109
Configuring a Crypto LUN	110
Removing a LUN from a CryptoTarget container	111
Crypto LUN parameters and policies	112
Modifying Crypto LUN parameters	114
LUN modification considerations	114
Force-enabling a disabled disk LUN for encryption	115
Configuring a tape LUN	115
Modify example	116
Configuring a multi-path Crypto LUN	117
Multi-path LUN configuration example	117
Tape pool configuration	120
Tape pool labeling	121
Creating a tape pool	123
Deleting a tape pool	124
Modifying a tape pool	124
Impact of tape LUN configuration changes	124
Impact of tape pool configuration changes	125
Data re-keying	125
Resource Allocation	125
Re-keying modes	126
Configuring a LUN for automatic re-keying	126
Initiating a manual re-key session	127
Suspension and resumption of re-keying operations	128
First time encryption	129
Resource allocation	129
First time encryption modes	129
Configuring a LUN for first time encryption	129

Chapter 4

Deployment Scenarios

In this chapter	131
Single encryption switch, two paths from host to target	132
Single fabric deployment - HA cluster	133
Single fabric deployment - DEK cluster	134
Dual fabric deployment - HA and DEK cluster	135
Multiple paths, one DEK cluster, and two HA clusters	136
Multiple paths, DEK cluster, no HA cluster	138

Deployment in Fibre Channel routed fabrics	139
Deployment as part of an edge fabric	141
Deployment with FCIP extension switches	142
Data mirroring deployment	143
If metadata is not present on the LUN	144
VmWare ESX server deployments	145

Chapter 5 Best Practices and Special Topics

In this chapter	147
Firmware download considerations	148
Firmware Upgrades and Downgrades	148
Specific guidelines and procedures	149
Configuration upload and download considerations	150
Configuration Upload at an encryption group leader node	150
Configuration upload at an encryption group member node	150
Information not included in an upload	150
Steps before configuration download	151
Configuration download at the encryption group leader	151
Configuration download at an encryption group member	151
Steps after configuration download	152
HP-UX considerations	153
Enable of a disabled LUN	153
Disk metadata	153
Tape metadata	153
Tape data compression	154
Tape pools	154
Tape block zero handling	154
Tape key expiry	155
DF compatibility for tapes	155
DF compatibility for disk LUNs	155
Configuring CryptoTarget containers and LUNs	156
Redirection zones	157
Deployment with Admin Domains (AD)	157
Master key usage in RKM and SKM environments	157
Do not use DHCP for IP interfaces	157
Ensure uniform licensing in HA clusters	157
Tape library media changer considerations	158
Turn off host-based encryption	158
Avoid double encryption	158
PID failover	158

Turn off compression on extension switches	158
Re-keying best practices and policies	159
Manual re-key	159
Latency in re-key operations	159
Allow re-key to complete before deleting a container	159
Re-key operations and firmware upgrades	159
Do not change LUN configuration while re-keying	160
Brocade native mode in LKM installations	160
Recommendation for Host I/O traffic during online rekeying and first time encryption	160
Changing IP addresses in encryption groups	160
Disabling the encryption engine	160
Recommendations for Initiator Fan-Ins	161
Best practices for host clusters in an encryption environment ...	162
HA Cluster Deployment Considerations and Best Practices	162

Chapter 6

Maintenance and Troubleshooting

In this Chapter	163
Encryption group and HA cluster maintenance	163
Removing a node from an encryption group	163
Deleting an encryption group	165
Removing an HA cluster member	166
Displaying the HA cluster configuration	166
Replacing an HA cluster member	167
Deleting an HA cluster member	170
Performing a manual failback of an encryption engine	170
Encryption group merge and split use cases	171
Configuration impact of encryption group split or node isolation	176
General encryption troubleshooting using the CLI	177
Troubleshooting examples using the CLI	179
Encryption Enabled Crypto Target LUN	179
Encryption Disabled Crypto Target LUN	180
Management application encryption wizard troubleshooting ...	181
Errors related to adding a switch to an existing group	181
Errors related to adding a switch to a new group	182
General errors related to the Configure Switch Encryption wizard	184
LUN policy troubleshooting	185
Loss of encryption group leader after power outage	186
MPIO and internal LUN states	187
Suspension and resumption of re-keying operations	187

Appendix A	State and Status Information	
	In this appendix	189
	Encryption engine security processor (SP) states	189
	Security processor KEK status	190
	Encrypted LUN states	190
Appendix B	LUN Policies	
	In this appendix	195
	DF-compatibility support for disk LUNs	195
	DF-compatibility support for tape LUNs	199
Appendix C	NS-Based Transparent Frame Redirection	
Appendix D	Supported Key Management Systems	
	In this appendix	203
	Key management systems	203
	The NetApp Lifetime Key Manager	204
	The NetApp DataFort Management Console	204
	Obtaining and importing the LKM certificate	205
	Registering the certificates	206
	Establishing the trusted link	208
	LKM key vault high availability deployment	209
	The RSA Key Manager	212
	Obtaining and Importing the RKM certificate	212
	Exporting the KAC certificate signing request (CSR)	212
	Submitting the CSR to a certificate authority	213
	Importing the signed KAC certificate	213
	Uploading the KAC and CA certificates onto the RKM appliance	215
	RKM key vault high availability deployment	216

The HP Secure Key Manager	218
Obtaining a signed certificate from the HP SKM appliance software	219
Importing a signed certificate	220
Exporting the KAC certificate request	221
Configuring a Brocade group.	221
Registering the Brocade user name and password in encryption groups.	222
Setting up the local certificate authority	222
Adding the local CA to the trusted CAs list.	223
Adding a server certificate for the SKM appliance	224
Downloading the local CA certificate file	225
Creating an SKM Key vault High Availability cluster	226
Copying the local CA certificate.	226
Adding an HP SKM appliance to a cluster	226
Signing the KAC certificate	227
Importing a signed certificate (SAN Management program) ..	229
SKM key vault high availability deployment.	229
Thales Encryption Manager for Storage	232
Generating the Brocade user name and password.	232
Adding a client	232
Signing the CSR	233
Registering the certificates	235
Thales key vault high availability deployment	235

Index

About This Document

In this chapter

- [How this document is organized](#) xiii
- [Supported hardware and software](#)..... xiv
- [What's new in this document](#)..... xiv
- [Document conventions](#) xiv
- [Notice to the reader](#) xvi
- [Additional information](#)..... xvi
- [Getting technical help](#)..... xvii
- [Document feedback](#) xviii

How this document is organized

- This document is organized to help you find the information that you want as quickly and easily as possible.

The document contains the following components:

- [Chapter 1, “Encryption overview,”](#) provides a task matrix, an overview of the data encryption switch and the encryption solution, and the terminology used in this document.
- [Chapter 2, “Encryption configuration using the Management application,”](#) describes how to configure and manage encryption features using DCFM.
- [Chapter 3, “Encryption configuration using the CLI,”](#) describes how to configure and manage encryption features using the command line interface.
- [Chapter 4, “Deployment Scenarios”](#) describes SAN configurations in which encryption may be deployed.
- [Chapter 5, “Best Practices and Special Topics,”](#) summarizes best practices and addresses special topics relevant to the implementation of encryption features.
- [Chapter 6, “Maintenance and Troubleshooting,”](#) provides information on troubleshooting and the most common commands and procedures to use to diagnose and recover from problems.
- [Appendix A, “State and Status Information,”](#) lists the encryption engine security processor (SP) states, security processor key encryption key (KEK) status information, and encrypted LUN states.
- [Appendix B, “LUN Policies,”](#) provides a DataFort compatibility support matrix for disk and tape LUNs, and includes LUN policy troubleshooting information.
- [Appendix C, “NS-Based Transparent Frame Redirection,”](#) provides a name server (NS)-based transparent frame redirection interop matrix.

- [Appendix D, “Supported Key Management Systems,”](#) describes supported key management systems, and provides procedures for certificate exchanges to enable mutual authentication of encryption switches or blades and key management appliances.

Supported hardware and software

- The following hardware platforms support data encryption as described in this manual.
 - Brocade DCX and DCX-4S with an FS8-18 encryption blade.
 - Brocade Encryption Switch.

What’s new in this document

Information about authentication cards, system cards, support of VmWare deployment scenarios, and the Thales Encryption Manager for Storage (TEMS) is included in this document. Note that TEMS is referred to as nCipher Key Authority (NCKA) in some sections of this document. Also, this document includes new information about support for multiple paths between hosts and targets, and key management appliance clusters.

Document conventions

This section describes text formatting conventions and important notice formats used in this document.

Text formatting

The narrative-text formatting conventions that are used are as follows:

bold text	Identifies command names Identifies the names of user-manipulated GUI elements Identifies keywords and operands Identifies text to enter at the GUI or CLI
<i>italic text</i>	Provides emphasis Identifies variables Identifies paths and Internet addresses Identifies document titles
<code>code text</code>	Identifies CLI output Identifies command syntax examples

For readability, command names in the narrative portions of this guide are presented in mixed lettercase: for example, switchShow. In actual examples, command lettercase is often all lowercase. Otherwise, this manual specifically notes those cases in which a command is case sensitive.

Command syntax conventions

Command syntax in this manual follows these conventions:

command	Commands are printed in bold.
--option, option	Command options are printed in bold.
-argument, arg	Arguments.
[]	Optional element.
<i>variable</i>	Variables are printed in italics. In the help pages, variables are <u>underlined</u> or enclosed in angled brackets < >.
...	Repeat the previous element, for example “member[:member...]”
value	Fixed values following arguments are printed in plain font. For example, --show WWN
	Boolean. Elements are exclusive. Example: --show -mode egress ingress
\	Backslash. Indicates that the line continues through the line break. For command line input, type the entire line without the backslash.

Notes, cautions, and warnings

The following notices and statements are used in this manual. They are listed below in order of increasing severity of potential hazards.

NOTE

A note provides a tip, guidance or advice, emphasizes important information, or provides a reference to related information.

ATTENTION

An Attention statement indicates potential damage to hardware or data.



CAUTION

A Caution statement alerts you to situations that can cause damage to hardware, firmware, software, or data.



DANGER

A Danger statement indicates conditions or situations that can be potentially lethal or extremely hazardous to you. Safety labels are also attached directly to products to warn of these conditions or situations.

Key terms

For definitions specific to Brocade and Fibre Channel, see the technical glossaries on Brocade Connect. See “[Brocade resources](#)” on page xvi for instructions on accessing Brocade Connect.

For definitions specific to this document, see “Terminology” on page 3.

For definitions of SAN-specific terms, visit the Storage Networking Industry Association online dictionary at:

<http://www.snia.org/education/dictionary>

Notice to the reader

This document may contain references to the trademarks of the following corporations. These trademarks are the properties of their respective companies and corporations.

These references are made for informational purposes only.

Corporation	Referenced Trademarks and Products
Microsoft Corporation	Windows, Windows NT, Internet Explorer
Net App	Lifetime Key Manager (LKM)
EMC	RSA Key Manager (RKM)
Hewlett Packard	Secure Key Manager (SKM)
IBM	IBM Tivoli Storage Manager 5.4 (Windows 2003)— Tape backup only, no support for tape pool
EMC Legato	Legato Networker 7.4 (Windows 2003 and Red Hat Linux 5.1)
Symantec	Symantec Veritas NetBackup 6.5 Enterprise Server (Windows 2003 and Solaris 10)
CommVault	Commvault Galaxy Data Protection 7.0 (Windows 2003)

Additional information

This section lists additional Brocade and industry-specific documentation that you might find helpful.

Brocade resources

To get up-to-the-minute information, go to <http://my.brocade.com> and register at no cost for a user ID and password.

For practical discussions about SAN design, implementation, and maintenance, you can obtain *Building SANs with Brocade Fabric Switches* through:

<http://www.amazon.com>

For additional Brocade documentation, visit the Brocade SAN Info Center and click the Resource Library location:

<http://www.brocade.com>

Release notes are available on the Brocade Connect Web site and are also bundled with the Fabric OS firmware.

Other industry resources

- White papers, online demos, and data sheets are available through the Brocade Web site at <http://www.brocade.com/products-solutions/products/index.page>.
- Best practice guides, white papers, data sheets, and other documentation is available through the Brocade Partner Web site.

For additional resource information, visit the Technical Committee T11 Web site. This Web site provides interface standards for high-performance and mass storage applications for Fibre Channel, storage management, and other applications:

<http://www.t11.org>

For information about the Fibre Channel industry, visit the Fibre Channel Industry Association Web site:

<http://www.fibrechannel.org>

Getting technical help

Contact your switch support supplier for hardware, firmware, and software support, including product repairs and part ordering. To expedite your call, have the following information available:

1. General Information

- Switch model
- Switch operating system version
- Error numbers and messages received
- **supportSave** command output
- Detailed description of the problem, including the switch or fabric behavior immediately following the problem, and specific questions
- Description of any troubleshooting steps already performed and the results
- Serial console and Telnet session logs
- syslog message logs

2. Switch Serial Number

The switch serial number and corresponding bar code are provided on the serial number label, as illustrated below.:



The serial number label is located as follows:

- *Brocade Encryption Switch*—On the switch ID pull-out tab located inside the chassis on the port side of the switch on the left.
- *Brocade DCX*—On the bottom right on the port side of the chassis
- *Brocade DCX-4S*—On the bottom right on the port side of the chassis, directly above the cable management comb.

3. World Wide Name (WWN)

Use the **licenseIdShow** command to display the WWN of the chassis.

If you cannot use the **licenseIdShow** command because the switch is inoperable, you can get the WWN from the same place as the serial number, except for the Brocade DCX. For the Brocade DCX, access the numbers on the WWN cards by removing the Brocade logo plate at the top of the non-port side of the chassis.

Document feedback

Quality is our first concern at Brocade and we have made every effort to ensure the accuracy and completeness of this document. However, if you find an error or an omission, or you think that a topic needs further development, we want to hear from you. Forward your feedback to:

documentation@brocade.com

Provide the title and version number of the document and as much detail as possible about your comment, including the topic heading and page number and your suggestions for improvement.

Encryption overview

In this chapter

• Host and LUN considerations	1
• Encryption configuration tasks	2
• Terminology	3
• The Brocade encryption switch	5
• The FS8-18 blade	6
• Performance licensing	6
• Recommendation for connectivity	7
• Usage limitations	7
• Brocade encryption solution overview	8
• Data encryption key life cycle management	10
• Key management systems	12
• Encryption switch initialization	13
• Support for Virtual Fabrics	13

Host and LUN considerations

Encrypting data-at-rest provides peace of mind in terms of protecting data from loss or theft, but very careful planning must be done to ensure encrypted data is handled correctly. Much of the planning must come from careful evaluation of host application and LUN resources, and of the path that the data will take to get from one or more hosts to a LUN.



CAUTION

When implementing encryption for data-at-rest, all hosts that access a LUN that is to hold encrypted data need to be configured for encryption to avoid data corruption. If a host, possibly in another fabric, writes cleartext to an encrypted LUN, the data on the LUN will be lost. The user must ensure that all hosts that can access a LUN are configured in the same manner.

1 Encryption configuration tasks

Encryption configuration tasks

Table 1 provides a high level overview and checklist of encryption configuration tasks. These tasks must be done in the order presented in the table. If the tasks are done out of order, unexpected errors may be encountered, and the results may be unpredictable. Some tasks can be done only at the command line interface (CLI). Other tasks may be done at the CLI, or at the Data Center Fabric Manager (DCFM) management program.

TABLE 1 High-level encryption configuration checklist

	Configuration task	For more information
<input type="checkbox"/>	Initialize the switch	“Initializing an encryption switch” on page 90
<input type="checkbox"/>	Configure the encryption group leader	“Basic encryption group configuration” on page 95 (CLI) <ul style="list-style-type: none"> • Creating an encryption group • Group-wide policy configuration
<input type="checkbox"/>	Set up and configure key vaults and register the key vaults with the encryption group leader.	Appendix D, “Supported Key Management Systems” <ul style="list-style-type: none"> • The NetApp Lifetime Key Manager (LKM) • The RSA Key Manager (RKM) • The HP Secure Key Manager (SKM) • The Thales Encryption Manager for Storage (TEMS, a.k.a., NCKA)
<input type="checkbox"/>	Add in all encryption group members and configure with key vaults if necessary.	“Adding a member node to an encryption group” on page 96 (CLI)
<input type="checkbox"/>	Create all HA Clusters, the members of which should span nodes.	“Master keys” on page 67 “High Availability (HA) cluster configuration” on page 100 (CLI)
<input type="checkbox"/>	Add in all CryptoTarget containers.	“CryptoTarget container configuration” on page 102 (CLI) <ul style="list-style-type: none"> • Frame redirection • Create a host - initiator zone • Creating a CryptoTarget container • Removing an initiator from a CryptoTarget container • Deleting a CryptoTarget container • Moving a CryptoTarget container
<input type="checkbox"/>	Create tape pools, if necessary.	“Adding tape pools” on page 33 (DCFM) “Tape pool configuration” on page 120 (CLI)
<input type="checkbox"/>	Configure all LUNs on all available paths	“Crypto LUN configuration” on page 109 (CLI) <ul style="list-style-type: none"> • Discovering a LUN • Configuring a Crypto LUN • Removing a LUN from a CryptoTarget container • Crypto LUN parameters and policies • Modifying Crypto LUN parameters • Force-enabling a disabled LUN for encryption • LUN configuration considerations • Configuring a tape LUN

Terminology

The following are definitions of terms used extensively in this document.

ciphertext	Encrypted data.
cleartext	Unencrypted data.
CryptoModule	The secure part of an encryption engine that is protected to the FIPS 140-2 level 3 standard. The term CryptoModule is used primarily in the context of FIPS authentication.
Data Encryption Key (DEK)	An encryption key generated by the encryption engine. The DEK is used to encrypt cleartext received from a host before it is sent to a target LUN, and to decrypt that data when it is retrieved by the host.
Data Encryption Key Cluster (DEK Cluster)	A cluster of encryption engines which can host all paths to a LUN and share the same data encryption key (DEK) set. The encryption engines can be in the same or different fabrics. DEK clusters enable host MPIO failover.
Encryption Engine	The entity within a node that performs encryption operations, including the generation of Data Encryption Keys.
Encryption Group	A collection of one or more DEK clusters, HA clusters, or both, which share the same key vault and device configuration, and is managed as a single group.
Failback	In the context of this implementation of encryption, failback refers to behavior after a failed encryption switch recovers. Devices that were transferred to another switch by failover processing may automatically be transferred back, or they may be manually switched back. This is determined as a configuration option.
Failover	In the context of this implementation of encryption, failover refers to the automatic transfer of devices hosted by one encryption switch to another encryption switch within a high availability cluster (HA cluster).
Group Leader	A group leader is a special node within an encryption group which acts as a group and cluster manager, and manages and distributes all group-wide and cluster-wide configurations to all members of the group or cluster.
High Availability Cluster (HA Cluster)	A collection of peer-level encryption engines that provide failover capabilities within a fabric.
Key Encryption Key	A key used to encrypt and decrypt Data Encryption Keys (DEKs) within encryption devices so that DEKs are transmitted in a secure manner outside of the encryption engines, and stored persistently inside key vaults.
Link Key	A shared secret exchanged between an encryption engine and a FIPS 140-2 level 3 certified key management appliance and key vault. The link key is a Key Encryption Key (KEK) that is used to encrypt Data Encryption Keys (DEKs) in transit over a secure connection to and from the key vault. The key management appliance decrypts the DEKs and stores them encrypted with its own master key.
Master Key	An Key Encryption Key (KEK) used to encrypt and decrypt DEKs when storing DEKs in opaque key vaults. There is one master key per encryption group. That means all node encryption engines within an encryption group use the same master key to encrypt and decrypt the DEKs.
Node	In terms of encryption, a switch, DCX, or DCX-4S through which users can manage an encryption engine.
Opaque Key Vault	A storage location that provides untrusted key management functionality. Its contents may be visible to a third party. DEKs in an opaque key vault are stored encrypted in a master key to protect them.

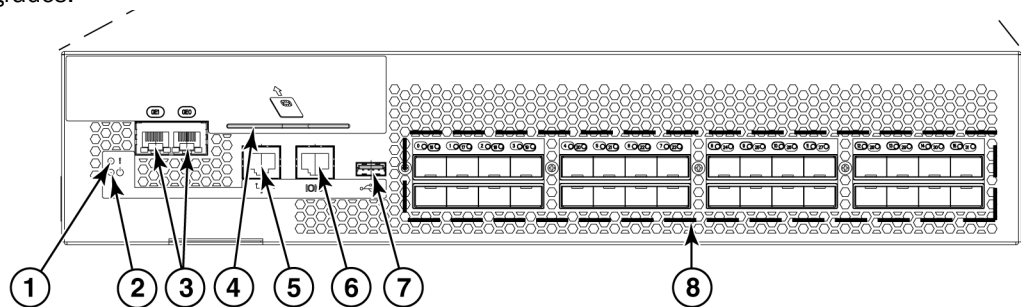
1 Terminology

Recovery cards	A set of smart cards that contain a backup master key. Each recovery card holds a portion of the master key. The cards must be gathered and read together from a card reader attached to a PC running the Brocade SAN Management Application to restore the master key. Recovery cards may be stored in different locations, making it very difficult to steal the master key. The cards should not be stored together, as that defeats the purpose.
Redirection zone	When encryption is implemented, data traffic is routed to and from virtual initiators and virtual targets. Redirection zones are automatically created to enable frame redirection to the virtual initiators and virtual targets.
Re-keying	Re-keying refers to decrypting data with the current Data Encryption Key (DEK), and encrypting it with a new DEK. This is done when the security of the current key is compromised, or when a DEK is configured to expire in a specific time frame. The re-keying operation can be used to encrypt existing data currently stored as cleartext. In that case, there is no existing DEK, and the data does not have to be decrypted before it is encrypted using the new DEK.
Trusted Key Vault	Very secure storage on a hardware appliance that establishes a trusted link with the encryption device for secure exchange of DEKs. DEKs are encrypted with the link for transit between the encryption device and the hardware appliance. At the hardware appliance, the DEKs are re-encrypted, using master key created and maintained by hardware appliance, and then stored in the trusted key vault.
Virtual Initiator	A logical entity that acts as a stand-in for a physical host when communicating with a physical target LUN.
Virtual Target	A logical entity that acts as a stand-in for a physical target LUN when communicating with a physical host. A virtual target is mapped one to one to a specific physical target.

The Brocade encryption switch

The Brocade encryption switch ([Figure 1](#)) is a high performance 32 port auto-sensing 8 Gbps Fibre Channel switch with data cryptographic (encryption/decryption) and data compression capabilities. The switch is a network-based solution that secures data-at-rest for heterogeneous tape drives, disk array LUNs, and virtual tape libraries by encrypting the data, using Advanced Encryption Standard (AES) 256-bit algorithms. Encryption and decryption engines provide in-line encryption services with up to 96 Gbps throughput for disk I/O (mix of ciphertext and cleartext traffic) and up to 48 Gbps throughput for tape I/O (mix of ciphertext and cleartext traffic). Refer to [“The FS8-18 blade”](#) on page 6 for information about license requirements for 48 Gbps and 96 Gbps bandwidth.

In addition to its 32 Fibre Channel ports, the switch has one RJ45 Gigabit Ethernet (GE) management port, two RJ45 GE ports for clustering interconnection and re-key synchronization, one RJ45 Serial console port, and one USB port for serviceability, error logging, and firmware upgrades.



- 1 Power LED.
- 2 Status LED.
- 3 RJ45 gigabit Ethernet ports for clustering and centralized management of multiple encryption switches through a group leader.
- 4 Smart card reader.
- 5 RJ45 gigabit Ethernet port for the management interface. This interface is used for the secure connection to the key vault location and to the Data Center Fabric Manager (DCFM).
- 6 RJ45 serial console port.
- 7 USB port for firmware upgrades and other support services.
- 8 Fibre Channel ports (0-31) - 1, 2, 4, or 8 Gbps auto-sensing F, FL, E, EX, or M ports to connect host servers, SAN disks, SAN tapes, edge switches, or core switches.

FIGURE 1 Brocade encryption switch

The FS8-18 blade

The FS8-18 blade provides the same features and functionality as the encryption switch. The FS8-18 blade installs on the Brocade DCX and DCX-4S. Four FS8-18 blades may be installed in a single DCX or DCX-4S.

Performance licensing

Encryption processing power is scalable, and may be increased by purchasing and installing an encryption performance license. The base unit Brocade Encryption Switch and FS8-18 Encryption Blade have a standard capacity of 48 Gbps of encryption processing power. Additional encryption processing power can be added for disk I/O by purchasing and installing a Disk Advanced Encryption Performance license. When the performance upgrade license is applied, encryption processing power of up to 96 Gbps is available. Note that when the license is applied to a DCX or DCX-4S chassis, it applies to all FS8-18 blades installed on that chassis.

Adding a license

The encryption performance licenses are added just like any other Fabric OS feature license. After the license is added, the encryption switch, DCX, or DCX-4S with encryption blades installed must be rebooted for the license to take effect. See the *Fabric OS Administrator's Guide* for information about obtaining and adding licenses.

Licensing best practices

Licenses installed on the switches and blades must have identical performance numbers when used together in high availability (HA) clusters or data encryption key (DEK) clusters.

Recommendation for connectivity

In order to achieve high performance and throughput, the encryption engines perform what is referred to as “cut-through” encryption. In simple terms this is achieved by encrypting the data in data frames on a per frame basis. This enables the encryption engine to buffer only a frame, encrypt it and send the frame out to the target on write I/Os. For read I/Os the reverse is done. This puts some constraints on the topology and the container configurations to support acceptable performance for encrypted and decrypted I/O to and from LUNs, and to support acceptable levels of scale in terms of the number of LUNs and the number of flows. The topology and container configuration constraint is stated below:

Care must be taken when connecting the encryption engines to the fabric and configuring crypto-target containers to be sure that the traffic flow between the host initiator and the physical storage array LUN through the container flows through only one encryption engine that is hosting the container. This is to avoid crisscrossing of flows to and from virtual entities; that is, from virtual targets and virtual initiators on two different encryption engines over the same path.

Although there is considerable flexibility in connecting and configuring the containers for encryption, the following guidelines are the recommended best practices:

- Host and Storage Array ports that are not involved in any encryption flow can be connected to any Encryption Engines.
- Recommendations for host and target ports with respect to encryption flows are as follows:
 - Only ISLs are connected to the Brocade Encryption Switch encryption engine in order to connect it to the fabric. No devices (initiators and targets) are connected to it.
 - Only host ports are connected to the FS8-18 blade encryption engine. and no ISLs are connected to it.

Usage limitations

There are usage limitations to be aware of when planning an encryption implementation:

- You cannot host both disk storage or tape storage on the same encryption switch or blade.
- Special redirection zones are created to handle data that is redirected to an encryption switch or blade. Quality of Service (QoS) cannot be applied to a redirection zone.
- In order for frame redirection to be applied, regular zones for hosts and targets must be defined in the effective configuration. Hosts and targets must be zoned together by worldwide port name (WWPN) rather than worldwide node name (WWNN) in configurations where frame redirection will be used. If hosts or targets are zoned together using worldwide node name, frame redirection will not occur properly.
- On tapes written in DataFort format, the encryption switch or blade cannot read and decrypt files with a block size of one MB or greater.
- The Top Talker feature is not compatible with redirection zones. The Top Talker feature should not be enabled when an encryption switch or blade is present in the fabric.

Brocade encryption solution overview

The loss of stored private data, trade secrets, intellectual properties, and other sensitive information through theft or accidental loss of disk or tape media can have widespread negative consequences for governments, businesses, and individuals. This threat is countered by an increasing demand from governments and businesses for solutions that create and enforce policies and procedures that protect stored data. Encryption is a powerful tool for data protection. Brocade provides an encryption solution that resides in a Storage Area Network (SAN) fabric. This location, between computers and storage, is ideal for implementing a solution that works transparently with heterogeneous servers, disk storage subsystems, and tape libraries. Data entering the SAN from a server is encrypted before it is written to storage. When stored data is encrypted, theft or loss of storage media does not pose a security threat.

Figure 2 provides a high level view of the Brocade encryption solution. Cleartext is sent from the server to the encryption engine, where it is encrypted into ciphertext using one of two encryption algorithms, one for disk storage targets, and one for tape storage targets. The encrypted data cannot be read without first being decrypted. The key management system is required for management of the data encryption keys (DEKs) that are generated by the encryption engine, and used for encrypting and decrypting the data. The key management system is provided by a third party vendor.

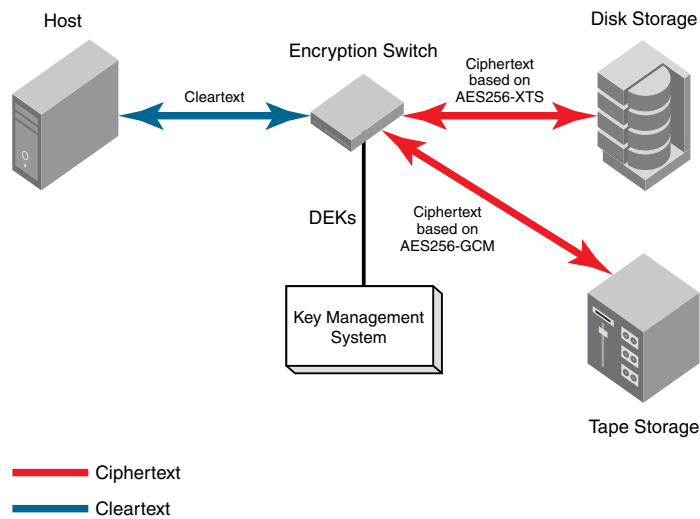


FIGURE 2 Encryption overview

Data flow from server to storage

The Brocade encryption switch can be introduced into a SAN with minimum disruption, with no need for SAN reconfiguration, and with no need to reconfigure host applications. Frames sent from a host and a target LUN are redirected to a virtual target associated with the encryption switch. The encryption switch then acts as a virtual initiator to forward the frames to the target LUN.

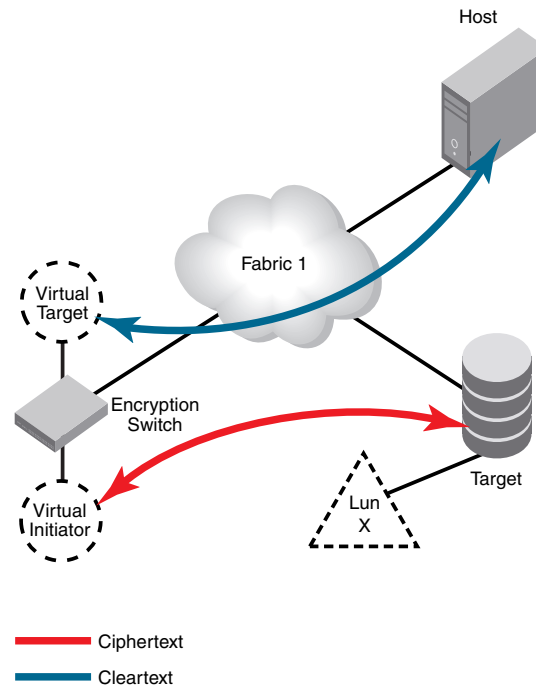


FIGURE 3 Frame redirection

Data encryption key life cycle management

Data encryption keys (DEKs) are generated by the encryption engine. Data is encrypted and decrypted using the same DEK, so a DEK must be preserved at least long enough to decrypt the ciphertext that it created. The length of time data is stored before it is retrieved can vary greatly, and some data may be stored for years or decades before it is accessed. To be sure the data remains accessible, DEKs may also need to be stored for years or decades. Key management systems provide life cycle management for all DEKs created by the encryption engine. Key management systems are provided by third party vendors.

Figure 4 shows the relationship of the LAN connections to the key vault and between encryption nodes.

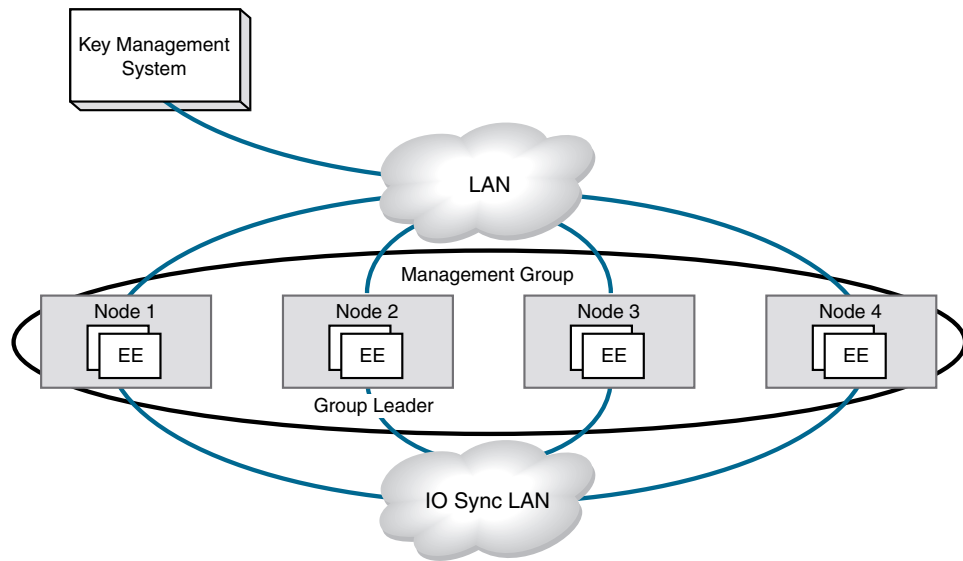


FIGURE 4 LAN connections to the key vault, and between encryption nodes

Regardless of the length of the life cycle, there are four stages in the life of a DEK, as shown in Figure 5. A DEK is created by an encryption engine, distributed, and stored in a key vault. The key is used to encrypt and decrypt data at least once, and possibly many times. A DEK may be configured to expire in a certain time frame, or it may become compromised. Under those conditions, it must be used one more time to decrypt the data, and the resulting cleartext is encrypted with a new key (re-keyed).

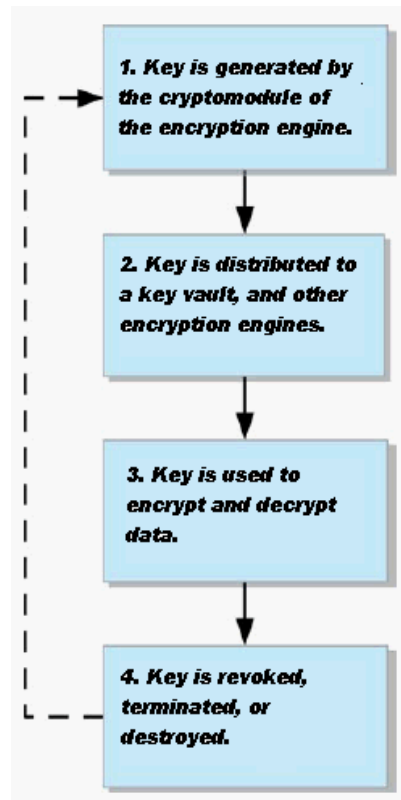


FIGURE 5 DEK life cycle

Key management systems

Key management systems are available from several vendors. This release supports the following leading key management systems:

- The NetApp Lifetime Key Manager (LKM) version 4.0 or later.
- The RSA Key Manager (RKM) version 2.1.3 or later, available through EMC.
- The HP Secure Key Manager (SKM) version 1.1 or later, available through Hewlett Packard.
- The Thales Encryption Manager for Storage (TEMS), sometimes referred to as NCKA within operational descriptions in this document.

Refer to [Appendix D, “Supported Key Management Systems”](#) for more information about supported key management systems.

Master key management

Communications with opaque key vaults are encrypted using a master key that is created by the encryption engine on the encryption switch. Currently, this includes the key vaults of all supported key management systems except NetApp LKM.

Master key generation

A master key must be generated by the group leader encryption engine. The master key can be generated once by the group leader, and propagated to the other members of an encryption group.

Master key backup

It is essential to back up the master key immediately after it is generated. The master key may be backed up to any of the following,

- To a file as an encrypted key.
- To the key management system as an encrypted key record.
- To a set of recovery smart cards. This option is only available if the switch is managed by the Data Center Fabric Manager (DFCM), and if a card reader is available for attachment to the DCFM workstation.

The use of smart cards provides the highest level of security. When smart cards are used, the key is split and written on up to five cards, and the cards may be kept and stored by up to five individuals, and all are needed to restore the master key.

Encryption switch initialization

Each encryption switch must be pre-initialized to be able to participate in a secure encryption environment. Pre-initialization establishes critical security parameters, such as certificates, and key pairs that are used to mutually authenticate each participating entity. Certificates and key pairs are needed to enable the following:

- Communication between the encryption engine and the switch control processor (CP).
- Communication between group leaders and nodes in an encryption group.
- Communication with key vaults.

Exporting, importing, and loading certificates

Certain certificates generated within an encryption switch or blade need to be exchanged with key vaults to enable mutual authentication. Refer to [Appendix D, “Supported Key Management Systems”](#) for information for each supported key vault.

Support for Virtual Fabrics

The Brocade encryption switch does not support the logical switch partitioning capability and can not be partitioned, but the switch can be connected to any Logical Switch partition or Logical Fabric using an E-Port.

The FS8-18 encryption blades are supported in only in a default switch partition All FS8-18 blades must be placed in a default switch partition in DCX or DCX-4S. The encryption resource from default switch partition/fabric can be shared with other logical switch partitions/fabrics or other fabrics only through external device sharing using FCR or EX_Ports through a base switch/fabric. A separate port blade must be used in the base switch/fabric for EX_Port connectivity from the logical switch partition (default switch partition) of FS8-18 blades and host/target fabrics. The EX_Port can be on any external FCR switch.

NOTE

Please refer to *Fabric OS Administrator’s Guide* for more details on how to configure the DCX and DCX-4S in virtual fabrics environments including configuration of default switch partition and any other logical switch partitions.

1 Support for Virtual Fabrics

Encryption configuration using the Management application

In this chapter

- Gathering information 16
- Encryption user privileges 17
- Encryption Center features 18
- Smart card usage 18
- Viewing and editing switch encryption properties 22
- Viewing and editing group properties 25
- Encryption Targets dialog box 34
- Creating a new encryption group 37
- Adding a switch to an encryption group 47
- Creating high availability (HA) clusters 50
- Adding encryption targets 54
- Configuring hosts for encryption targets 61
- Adding Target Disk LUNs for encryption 62
- Adding Target Tape LUNs for encryption 65
- Configuring encrypted storage in a multi-path environment 66
- Master keys 67
- Zeroizing an encryption engine 77
- Tracking Smart Cards 79

Gathering information

Before you use the encryption setup wizard for the first time, you should also have a detailed configuration plan in place and available for reference. The encryption setup wizard assumes the following:

- You have a plan in place to organize encryption devices into encryption groups.
- If you want redundancy and high availability in your implementation you have a plan to create high availability (HA) clusters of two encryption switches or blades to provide failover support.
- All switches in the planned encryption group are interconnected on an I/O synch LAN.
- The management ports on all encryption switches and DCX CPs that have encryption blades installed have a LAN connection to the SAN management program, and are available for discovery.
- A supported key management appliance is connected on the same LAN as the encryption switches, DCX CPs, and the SAN Management program.
- An external host is available on the LAN to facilitate certificate exchange.
- Switch KAC certificates have been signed by a Certificate Authority (CA), and stored in a known location.
- Key management system (key vault) certificates have been obtained and stored in a known location.

Encryption user privileges

In the Management application, resource groups are assigned privileges, roles, and fabrics. Privileges are not directly assigned to users; users get privileges because they belong to a role in a resource group. A user can only belong to one resource group at a time.

The Management application provides three pre-configured roles:

- Storage encryption configuration.
- Storage encryption key operations.
- Storage encryption security.

[Table 2](#) lists features and the associated roles with read/write access and read-only access.

TABLE 2 Role-based access control privileges and descriptions

Privilege	Read-Only	Read/Write
Storage Encryption Configuration	Disables all functions from the Encryption Center dialog box except view.	Enables the following functions from the Encryption Center dialog box: <ul style="list-style-type: none"> • Launch the Configure Encryption dialog. • View switch, group, or engine properties. • View the Encryption Group Properties Security tab. • View encryption targets, hosts, and LUNs. • Create a new encryption group or add a switch to an existing encryption group. • Edit group engine properties (except for the Security tab) • Add targets. • Select encryption targets and LUNs to be encrypted or edit LUN encryption settings. • Edit encryption target hosts configuration. • Change routing mode on an encryption engine.
Storage Encryption Key Operations	Disables all functions from the Encryption Center dialog box except view.	Enables the following functions from the Encryption Center dialog box: <ul style="list-style-type: none"> • Launch the Configure Encryption dialog. • View switch, group, or engine properties, • View the Encryption Group Properties Security tab. • View encryption targets, hosts, and LUNs. • Initiate manual LUN re-keying. • Enable and disable an encryption engine. • Zeroize an encryption engine. • Restore a master key. • Edit key vault credentials.
Storage Encryption Security	Disables all functions from the Encryption Center dialog box except view.	Enables the following functions from the Encryption Center dialog box: <ul style="list-style-type: none"> • Launch the Configure Encryption dialog. • View switch, group, or engine properties. • View encryption targets, hosts, and LUNs. • Create a master key. • Backup a master key. • Enable encryption functions after a power cycle. • View and modify settings on the Encryption Group Properties Security tab (quorum size, authentication cards list and system card requirement). • Establish link keys for LKM key managers.

Encryption Center features

The **Encryption Center** dialog box (Figure 6) is the single launching point for all encryption-related configuration in the Management application. It also provides a table that shows the general status of all encryption-related hardware and functions at a glance.

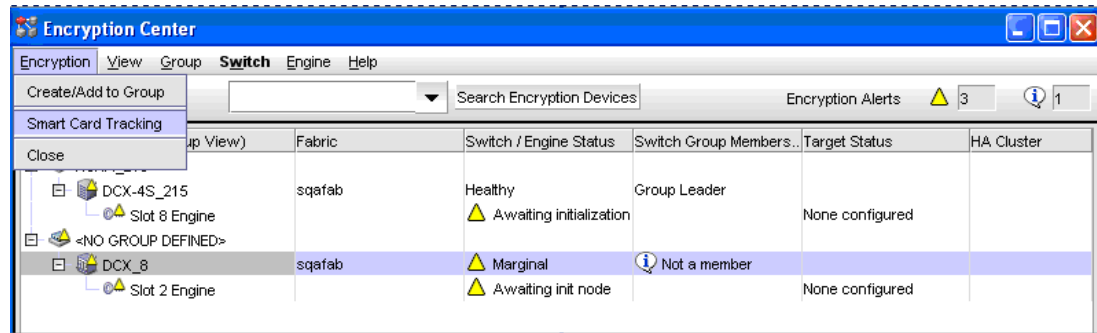


FIGURE 6 Encryption Center dialog box

The **Encryption Center** dialog box differs from the previous **Configure Encryption** dialog box. The buttons at the bottom of the dialog box are replaced with menus that are selected from a menu bar, or alternatively, by right-clicking an item in the table.

Smart card usage

Smart Cards are credit card-sized cards that contain a CPU and persistent memory. Smart cards can be used as security devices. With Brocade encryption switches, smart cards can be used to do the following:

- Control user access to the Management application security administrator roles.
- Control activation of encryption engines.
- Securely store backup copies of master keys.

Smart card readers provide plug-and-play interface to read and write to a smart card. The following smart card readers are supported:

- GemPlus GemPC USB
<http://www.gemalto.com/readers/index.html>
- SCM MicrosystemsSCR331
http://www.scmmicro.com/security/view_product_en.php?PID=2

See the following procedures for instructions about how to configure a Smart Card:

- “[Registering authentication cards from a card reader](#)” on page 19
- “[Registering system cards from a card reader](#)” on page 21
- “[Saving a master key to a smart card set](#)” on page 71
- “[Restoring a master key from a smart card set](#)” on page 75

Registering authentication cards from a card reader

When authentication cards are used, one or more authentication cards must be read by a card reader attached to a Management application PC to enable certain security sensitive operations. These include the following:

- Master key generation, backup, and restore operations.
- Replacement of authentication card certificates.
- Enabling and disabling the use of system cards.
- Changing the quorum size for authentication cards.
- Establishing a trusted link with the NetApp LKM key manager.

Authentication requires a quorum of authentication cards. The authentication provided by the quorum of authentication cards is given a lifespan of ten minutes, unless the authentication is explicitly cancelled, or the switch is rebooted or power-cycled. This prevents indefinite open-ended access to security sensitive operations after authentication. If the lifespan expires, and pending operations are allowed to complete, but new operations will require re-authentication.

To register an authentication card or a set of authentication cards from a card reader, have the cards physically available. Authentication cards can be registered during encryption group or member configuration when running the configuration wizard, or they can be registered using the following procedure.

1. Select **Configure > Encryption** from the menu bar.

The **Encryption Center** dialog box displays.

2. Select an encryption group, and select **Security Settings**.
3. Select the **Quorum Size**.

The quorum size is the minimum number of cards necessary to enable the card holders to perform the security sensitive operations listed above. The maximum quorum size is five cards. The actual number of authentication cards registered is always more than the quorum size, so if you set the quorum size to five, for example, you will need to register at least six cards in the subsequent steps.

NOTE

Ignore the **System Cards** setting. Refer to [“Enabling or disabling the system card requirement”](#) on page 21 for information on its usage.

4. Click **Next**.

The **Register Authentication Cards** dialog is displayed. This dialog include a table that shows all registered authentication cards.

5. Select **Register from Card Reader** to register a new card.

The **Add Authentication Card** dialog box is displayed.

6. Insert a smart card into the card reader. Be sure to wait for the card serial number to appear, and then enter card assignment information, as directed.
7. Click **OK**.
8. Wait for the confirmation dialog box indicating initialization is done, and click **OK**.

The card is added to the **Registered Authentication Cards** table on the **Authentication Cards** dialog box.

9. Repeat steps 7 through 10 until you have registered all the cards, and they all display in the **Registered Authentication Cards** table on the **Authentication Cards** dialog box. Remember that you need to register the number selected as the quorum size plus one.

Registering authentication cards from the database

Smart cards that are already in the Management program's database can be registered as authentication cards.

1. From the **Register Authentication Cards** dialog box, select **Register from Archive**.
The **Authentication Cards** dialog box displays, showing a list of smart cards in the database.
2. Select the card from the table, and click **OK**.
3. Wait for the confirmation dialog box indicating initialization is done, and click **OK**.
The card is added to the **Registered Authentication Cards** table.

De-registering an authentication card

Authentication cards can be removed from the database and the switch by de-registering them. Use the following procedure to de-register an authentication card.

1. Select the authentication card on the **Authentication Card** table.
2. Click **Deregister**.
3. A confirmation dialog box is displayed. Click **OK** to confirm de-registration.
The **Encryption Group** dialog box displays.
4. Click **OK** on the **Encryption Group** dialog box.
The card is de-registered from the group.

Using authentication cards

When a quorum of authentication cards are registered for use, an **Authenticate** dialog box is displayed to grant access to the following:

- The **Encryption Group Properties** dialog box **Security** tab.
- The **Encryption Group Properties** dialog box **Link Keys** tab.
- The **Master Key Backup** dialog box.
- The **Master Key Restore** dialog box.
- The **Edit System Card** dialog box.

To authenticate using a quorum of authentication cards, do the following:

1. When the **Authenticate** dialog box is displayed, gather the number of cards needed, as directed by instructions on the dialog box. The currently registered cards and the assigned owners are listed in the table near the bottom of the dialog box.
2. Insert a card, and wait for the ID to appear in the **Card ID** field.
3. Enter the assigned password.
4. Click **Authenticate**.

5. Wait for the confirmation dialog box, and click **OK**.
6. Repeat steps two through five for each card until the quorum is reached.
7. Click **OK**.

Registering system cards from a card reader

System cards are smart cards that can be used to control activation of encryption engines. Encryption switches and blades have a card reader that enables the use of a system card. System cards discourage theft of encryption switches or blades by requiring the use of a system card at the switch or blade to enable the encryption engine. When the switch or blade is powered off, the encryption engine will not work without first inserting a system card into its card reader. If someone removes a switch or blade with the intent of accessing the encryption engine, it will function as an ordinary FC switch or blade when it is powered up, but use of the encryption engine is denied.

To register a system card from a card reader, a smart card must physically available. System cards can be registered during encryption group or member configuration when running the configuration wizard, or they can be registered using the following procedure.

1. Select **Configure > Encryption** from the menu bar.
The **Encryption Center** dialog box displays.
2. Select the switch from the **Encryption Devices** table, and select **Switch > System Cards** from the menu task bar, or right-click the switch or and select **System Card**.
The **Register System Card** dialog box is displayed.
3. Insert a smart card into the card reader. Be sure to wait for the card serial number to appear, and then enter card assignment information, as directed.
4. Click **OK**.
5. Wait for the confirmation dialog box indicating initialization is done, and click **OK**.
The card is added to the **Registered System Cards** table on the **System Cards** dialog box.
6. Store the card in a secure location, not in the proximity of the switch or blade.

De-registering a system card

System cards can be removed from the database by de-registering them. Use the following procedure to de-register a system card.

1. From the **Register System Card** dialog box, select the system card you want to de-register.
2. Click **Deregister**.
3. A confirmation dialog box is displayed. Click **OK** to confirm de-registration.
The card is removed to the **Registered System Cards** table.

Enabling or disabling the system card requirement

If you want to use a system card to control activation of an encryption engine on a switch, you must enable the system card requirement. You can use the following procedure to enable or disable the system card requirement.

- **Node WWN** - the world wide name of the node.
- **Switch Status** - the health status of the switch. Possible values are Healthy, Marginal, Down, Unknown, Unmonitored, and Unreachable.
- **Switch Membership Status** - the alert or informational message description which details the health status of the switch. Possible values are Group Member, Leader-Member Comm, Error, Discovering, and Not a member.
- **Encryption Group** - the name of the encryption group to which the switch belongs.
- **Encryption Group Status** - Possible values are:
 - **OK - Converged** - the group leader can communicate with all members.
 - **Degraded** - the group leader cannot communicate with one or more members.
 - **Unknown** - the group leader is in an unmanaged fabric.

NOTE

When a group is in the **Degraded** state, the following operations are not allowed: key vault changes, master key operations, enable/disable encryption engines, Failback mode changes, HA Cluster creation or addition (removal is allowed), tape pool changes, and any configuration changes for storage targets, hosts, and LUNs.

- **Fabric** - the name of the fabric to which the switch belongs.
- **Domain ID** - the domain ID of the selected switch.
- **Firmware Version** - the current encryption firmware on the switch.
- **Primary Key Vault Link Key Status** - the possible statuses are as follows:
 - **Not Used** - the key vault type is not LKM.
 - **No Link Key** - no access request was sent to an LKM yet, or a previous request was not accepted.
 - **Waiting for LKM approval** - a request was sent to LKM and is waiting for the LKM administrator's approval.
 - **Waiting for local approval** - a response was received from LKM.
 - **Created, not validated** - the interim state until first used.
 - **OK** - a shared link key exists and has been successfully used.
- **Primary Key Vault Connection Status** - whether the primary key vault link is connected. Possible values are Unknown, Key Vault Not Configured, No Response, Failed authentication, and Connected.
- **Backup Key Vault Link Key Status** - the possible statuses are as follows:
 - **Not Used** - the key vault type is not LKM.
 - **No Link Key** - no access request was sent to an LKM yet, or a previous request was not accepted.
 - **Waiting for LKM approval** - a request was sent to LKM and is waiting for the LKM administrator's approval.
 - **Waiting for local approval** - a response was received from LKM.
 - **Created, not validated** - the interim state until first used.
 - **OK** - a shared link key exists and has been successfully used.

2 Viewing and editing switch encryption properties

- **Backup Key Vault Connection Status** - whether the backup key vault link is connected. Possible values are Unknown, Key Vault Not Configured, No Response, Failed authentication, and Connected.
- **Public Key Certificate** text box - the switch's KAC certificate, which must be installed on the primary and backup key vaults.
- **Save As** button - saves the certificate to a file in PEM format. The file may be loaded into the key vault using the key vault's tools.
- **Encryption Engine Properties** table - the properties for the encryption engine. There may be 0 to 4 slots, one for each encryption engine in the switch.
- **Current Status** - the status of the encryption engine. Common values are Not Available (not initialized), Disabled, Operational, need master/link key, and Online.
- **Set State To** - enter a new value, enabled or disabled, and click OK to apply the change.
- **Total Targets** - the number of the encrypted target device.
- **Routing Mode** - the routing mode of the encryption engine. Only Shared mode is supported.
- **HA Cluster Peer** - the name and location of the high-availability (HA) cluster peer (another encryption engine in the same group), if in an HA configuration.
- **HA Cluster Name** - the name of the High Availability (HA) cluster. The name can have a maximum of 31 characters. Letters, digits, and underscores are allowed.
- **Media Type** - the media type of the encryption engine. Possible values are Disk and Tape.
- **System Card** - the current status of system card information for the encryption engine. (registered or not registered).

Saving the public key certificate

To save the certificate to a file in PEM format, complete the following steps.

1. Click **Save As**.

The **Save** dialog box displays.

2. Browse to the location where you want to save the certificate.
3. Click **Save**.

You can now load the file into the key vault using the key vault's tools.

Enabling the encryption engine state

To enable the encryption engine state, complete the following steps.

1. Select **Enabled** from the **Set State To** list.
2. Click **OK**.

Disabling the encryption engine state

To disable the encryption engine state, complete the following steps.

1. Select **Disabled** from the **Set State To** list.
2. Click **OK**.

Viewing and editing group properties

To view encryption group properties, complete the following steps.

1. Select **Configure > Encryption**.

The **Encryption Center** dialog box displays.

2. If groups are not visible in the **Encryption Devices** table, select **View > Groups** from the menu bar.

The encryption groups display in the **Encryption Devices** table.

3. Select a group from the **Encryption Devices** table, and select **Group > Properties** from the menu bar, or right-click the group and select **Properties**.

The **Encryption Group Properties** dialog box, shown in [Figure 7](#), has six tabs which are defined in this section:

- “[General tab](#)” on page 26
- “[Members tab](#)” on page 26
- “[Security tab](#)” on page 29
- “[HA Clusters tab](#)” on page 29
- “[Engine Operations tab](#)” on page 30
- “[Link Keys tab](#)” on page 31
- “[Tape Pools tab](#)” on page 32

NOTE

The **Link Keys** tab appears only if the key vault type is NetApp LKM.

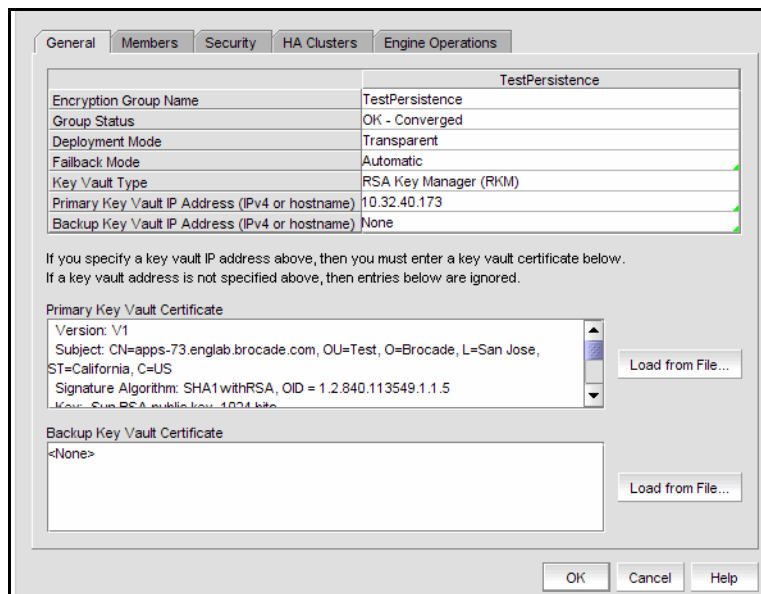


FIGURE 8 Encryption Group Properties dialog box

General tab

The properties displayed in the **General** tab are described below.

- **Encryption group name** - the name of the encryption group.
- **Group status** - the status of the encryption group, which can be **OK-Converged** or **Degraded**. Degraded means the group leader cannot contact all of the configured group members.
- **Deployment mode** - the group's deployment mode, which is transparent.
- **Failback mode** - The group's failback mode, which can be automatic or manual. For Fabric OS versions earlier than 6.2.0, the failback mode must be set manually using the CLI.
- **Key vault** - the vault type, either RSA Key Manager (RKM) NetApp Lifetime Key Manager (LKM), HP Secure Key Manager (SKM), or nCipher Key Authority (NCKA).
- **Primary key vault IP address** - The IP address of the primary key vault, either IPv4 or host name.
- **Backup key vault IP address** - the IP address of the backup key vault.
- **Primary key vault certificate** - the details of the primary vault certificate; for example, version and signature information.
- **Backup key vault certificate** - the details of the backup vault certificate; for example, version and signature information.

Members tab

The **Group Members** tab lists group switches, their role, and their connection status with the group leader. The tab displays the configured membership for the group (none of the table columns are editable). The list can be different from the members displayed in the **Encryption Center** dialog box if some configured members are unmanaged, missing, or in a different group.

Possible **Connection Status** values are as follows:

- **Group Leader** - this switch is the group leader so there is no connection status.
- **Trying to Contact** - the member is not responding to the group leader. This may occur if the member switch is not reachable by way of the management port, or if the member switch does not believe it is part of the encryption group.
- **Configuring** - the member switch has responded and the group leader is exchanging information. This is a transient condition that exists for a short time after a switch is added or restored to a group.
- **OK** - the member switch is responding to the group leader switch.
- **Not Available** - the group leader is not a managed switch, so connection statuses are not being collected from the group leader.

Members tab Remove button

You can click the **Remove** button to remove a selected switch or an encryption group from the encryption group table.

- You cannot remove the group leader unless it is the only switch in the group. If you remove the group leader, the Management application also removes the HA cluster, the target container, and the tape pool (if configured) that are associated with the switch.
- If you remove a switch from an encryption group, the Management application also removes the HA cluster and target container associated with the switch.

NOTE

If the encryption group is in a degraded state, the Management application does not remove the HA clusters or target containers associated with the switch. In this case, a pop-up error message displays.

- If you remove the last switch from a group, the Management application also deletes the group.

Consequences of removing an encryption switch

Table 3 explains the impact of removing switches.

TABLE 3 Switch removal warnings

Switch configuration	Impact of removal
The switch is the only switch in the encryption group.	The encryption group is also removed.
The switch has configured encryption targets on encryption engines.	<ul style="list-style-type: none"> • The switch is configured to encrypt traffic to one or more encryption targets. • The target container configuration is removed. • The encrypted data remains on the encryption target but is not usable until the encryption target is manually configured on another encryption switch.
The switch has encryption engines in HA Clusters.	The HA Clusters are removed. High availability is no longer provided to the other encryption engine in each HA Cluster.



CAUTION

The encryption target data is visible in encrypted format to zoned hosts. It is strongly recommended that you remove the encryption targets from all zones before you disable encryption. Otherwise, hosts may corrupt the encrypted data by writing directly to the encryption target without encryption.

Figure 9 shows the warning message that displays if you click **Remove** to remove a switch.

2 Viewing and editing group properties

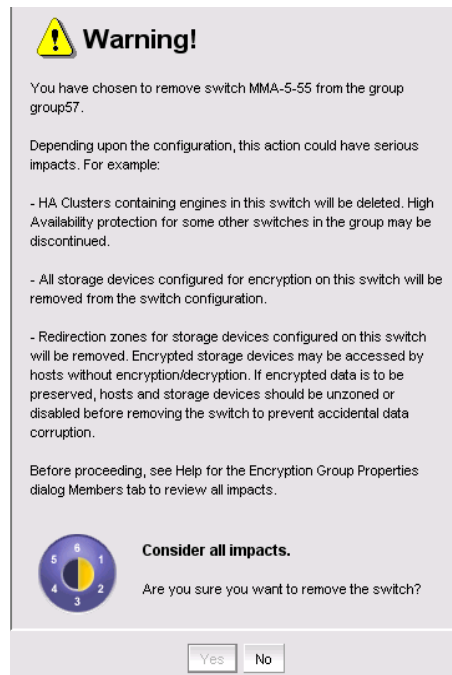


FIGURE 9 Removal of switch warning

Figure 10 shows the warning message that displays if you click **Remove** to remove an encryption group.

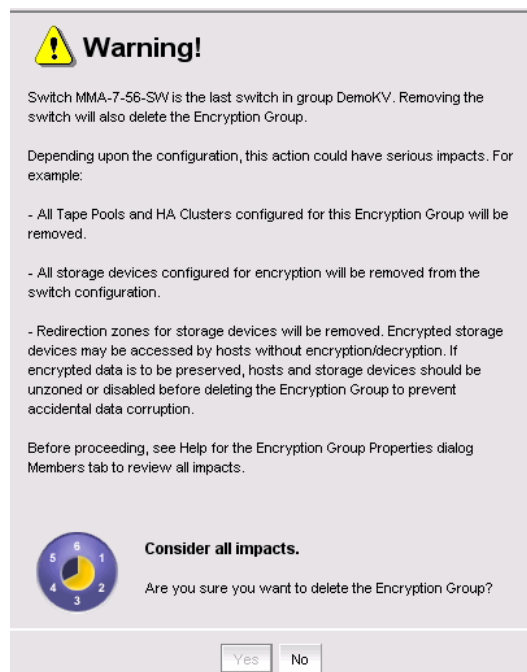


FIGURE 10 Removal of switch in encryption group warning

Security tab

The **Security** tab (Figure 11) displays the status of the master key for the encryption group.

NOTE

You must enable encryption engines before you back up or restore master keys.

Master key actions are as follows:

- **Back up a master key**, which is enabled any time a master key exists.
- **Restore a master key**, which is enabled when either no master key exists or the previous master key has been backed up.
- **Create a new master key**, which is enabled when no master key exists or the previous master key has been backed up.

See “[Master keys](#)” on page 67 for complete information about managing master keys.

NOTE

Encryption is not allowed until the master key has been backed up.

General Members **Security** HA Clusters Tape Pools Engine Operations

Master Key

When data encryption keys are stored in an opaque key vault, the data encryption keys are "wrapped" with a master key. Encryption is not allowed until the master key has been backed up.

Master Key Status: Created and backed up Master Key Actions ▼

None of the encryption engines in this encryption group have a copy of the master key. The master key should be restored from a backup.

System Cards

A system card is optional. When configured, they discourage hardware theft because they are needed at power-up in order to enable encryption.

Required Not Required

Authentication Cards

Authentication cards are optional. Changes to quorum size or authentication card registrations require approval of the current quorum upon clicking OK.

Authentication Card Quorum Size: 0 ▼

Registered Authentication Cards

Group Card #	Card ID	First Name	Last Name	Notes
1	qc.4250420d02048085	Peter	Long	
2	qc.4250420d0204647e	Tina	Eucerin	
3	qc.4250420d02047d7e			
4	qc.4250420d02045e77			

FIGURE 11 Encryption Group Properties - Security tab

HA Clusters tab

HA clusters are groups of encryption engines that provide high availability features. If one of the engines in the group fails or becomes unreachable, the other cluster member takes over the encryption and decryption tasks of the failed encryption engine. An HA cluster consists of exactly two encryption engines. See “[Creating high availability \(HA\) clusters](#)” on page 50.

The **HA Clusters** tab (Figure 12) allows you to create and delete HA clusters, add encryption engines to and remove encryption engines from HA clusters, and failback an engine.

2 Viewing and editing group properties

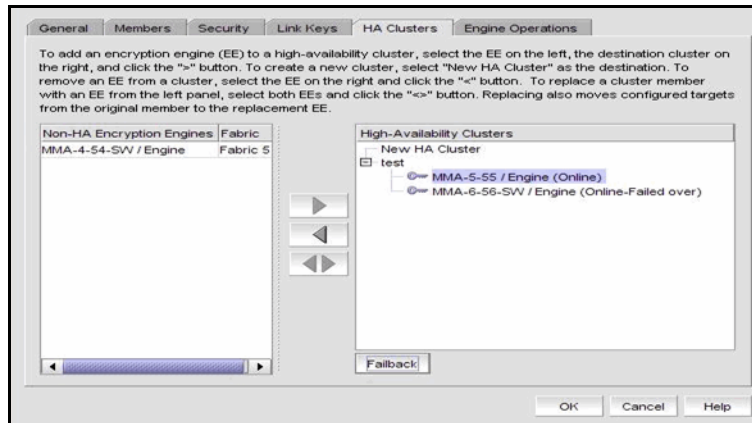


FIGURE 12 Encryption Group Properties - HA Clusters tab

Engine Operations tab

The **Engine Operations** tab (Figure 13) enables you to replace an encryption engine in an encryption switch with another encryption engine in another switch within a DEK Cluster environment. A DEK Cluster is a set of encryption engines that encrypt the same target storage device. DEK Clusters do not display in the Management application, they are an internal implementation feature and have no user-configurable properties.

NOTE

You cannot replace an encryption engine if it is part of an HA Cluster. For information about HA Clusters, refer to [“HA Clusters tab”](#) on page 29.

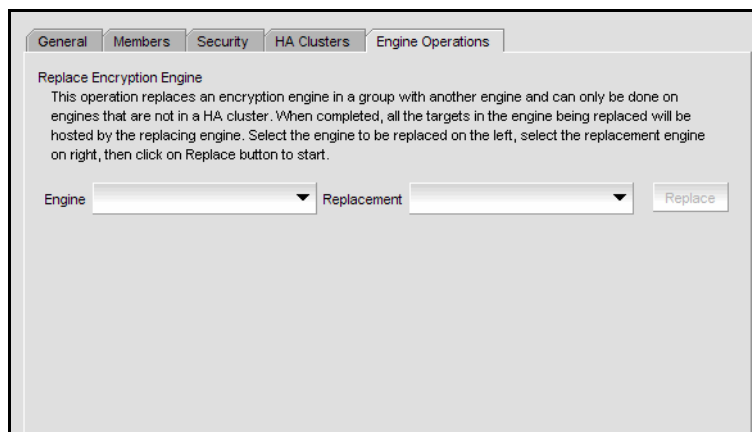


FIGURE 13 Encryption Group Properties - HA Clusters tab

Replacing an encryption engine

To replace an encryption engine in an encryption group with another encryption engine within a DEK Cluster, complete the following steps.

1. Select **Configure > Encryption**.
The **Encryption Center** dialog box displays.

2. If groups are not visible in the **Encryption Devices** table, select **View > Groups** from the menu bar.

The encryption groups display in the **Encryption Devices** table.

3. Select an encryption group from the tree, and select **Group > Properties** from the menu bar, or right-click the encryption group and select **Properties**.

The **Encryption Group Properties** dialog box displays.

4. Click the **Engine Operations** tab.
5. Select the engine you want to replace in the **Engine** list.
6. Select the engine you want to use as the replacement in the **Replacement** list.
7. Click **Replace**.

All containers hosted by the current engine (**Engine** list) are replaced by the new engine (**Replacement** list).

Link Keys tab

Connections between a switch and an NetApp LKM key vault require a shared link key. Link keys are used only with LKM key vaults. They are used to protect data encryption keys in transit to and from the key vault. There is a separate link key for each key vault for each switch. The link keys are configured for a switch but are stored in the encryption engines, and all the encryption engines in a group share the same link keys.

You must create link keys under the following circumstances:

- When a new encryption group is created.
- When a new switch is added to an encryption group.
- When a new key vault is added to an encryption group.
- After all encryption engines in a switch have been zeroized.
- When all of the encryption blades have been removed from a director and one or more new encryption blades have been added.

The **Link Keys** tab displays a table that shows link key status for each switch in an encryption group.

Tape Pools tab

Tape pools are managed from the **Tape Pools** tab.

Figure 14 displays the tape pools tab.

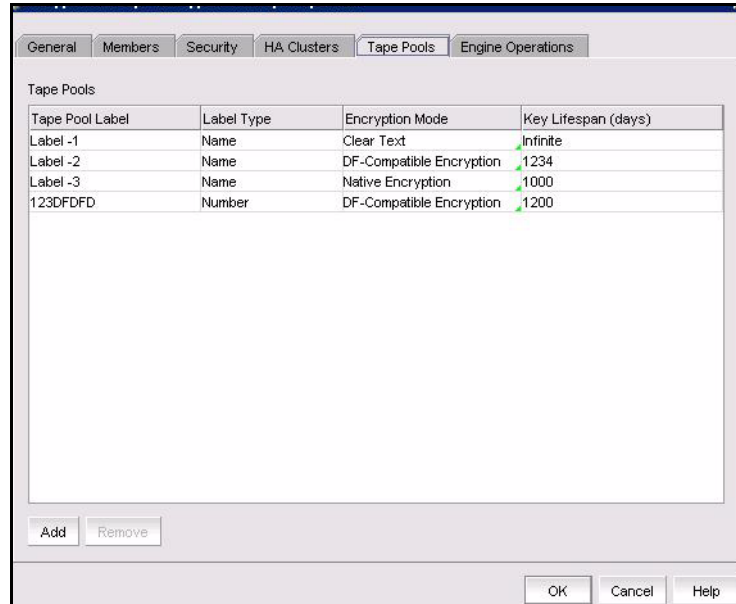


FIGURE 14 Encryption Group Properties - Tape Pools tab

- If you want to remove a tape pool, select one or more tape pools in the list and click **Remove**.
- To modify the tape pool, remove the entry and add a new tape pool. See [“Adding tape pools”](#) on page 33 for more information.

Tape pools overview

Tape cartridges and volumes may be organized into a tape pool (a collection of tape media). The same data encryption keys are used for all cartridges and volumes in the pool. Tape pools are used by backup application programs to group all the tape volumes used in a single backup or in a backup plan. The tape pool name or number used must be the same name or number used by the host backup application. If the same tape pool name or number is configured for an encryption group, tapes in that tape pool are encrypted according to the tape pool settings instead of the tape LUN settings.

Encryption switches and encryption blades support tape encryption at the tape pool level (for most backup applications) and at the LUN (tape drive) level. Since Tape Pool policies override the LUN (tape drive) policies, the LUN pool policies are used only if no tape pools exist, or if the tape media/volume does not belong to any configured tape pools.

All encryption engines in the encryption group share the tape pool definitions. Tapes can be encrypted by an encryption engine, where the container for the tape target LUN is hosted. The tape media is mounted on the tape target LUN.

Tape pool definitions are not needed to read a tape. Tape pool definitions are only used when writing to tape.

Adding tape pools

A tape pool can be identified by either a name or a number, but not both. Tape pool names and numbers must be unique within the encryption group. When a new encryption group is created, any existing tape pools in the switch are removed and must be added.

1. Select **Configure > Encryption** from the menu bar.

The **Encryption Center** dialog box displays.

2. If groups are not visible in the **Encryption Devices** table, select **View > Groups** from the menu bar.

The encryption groups display in the **Encryption Devices** table.

3. Select an encryption group from the tree, and select **Group > Tape Pools** from the menu bar, or right-click the encryption group and select **Tapepools**.

The **Add Tape Pool** dialog box displays. The **Name** tape pool label type is the default; however, you can change the tape pool label type to its number by selecting **Number**, shown in [Figure 16](#).

FIGURE 15 Add Tape Pool by name dialog box

FIGURE 16 Add Tape Pool by number dialog box

4. Specify the **Tape Pool Label Type**. Tape pools can be identified by either a name or a number, shown in [Figure 15](#) and [Figure 16](#).
5. Enter a name for the tape pool. If you selected **Number** as the **Tape Pool Label Type**, the name must match the tape pool label or tape ID/number that is configured on the tape backup/restore application.
6. Select the **Encryption Mode**.

Choices include **Clear Text**, **DF-Compatible Encryption**, and **Native Encryption**. **DF-Compatible Encryption** is valid only when LKM is the key vault. The **Key Lifespan (days)** field is editable only if the tape pool is encrypted. If **Clear Text** is selected as the encryption mode, the key lifespan is disabled.

NOTE

You cannot change the encryption mode after the tape pool I/O begins.

7. Enter the number of days that you want to use a key before obtaining a new key, if you want to enforce a key lifespan. The default is Infinite (a blank field or a value of 0) .

NOTE

The key lifespan interval represents the key expiry timeout period for tapes or tape pools. You can only enter the **Key Lifespan** field if the tape pool is encrypted. If **Clear Text** is selected as the encryption mode, the **Key Lifespan** field is disabled.

8. Click **OK**.

Encryption Targets dialog box

The **Encryption Targets** dialog box enables you to send outbound data that you want to store as ciphertext to an encryption device. The encryption target acts as a virtual target when receiving data from a host, and as a virtual initiator when writing the encrypted data to storage.

To access the Encryption Targets dialog box, complete the following steps.

1. Select **Configure > Encryption** from the menu bar.

The **Encryption Center** dialog box displays, showing the status of all encryption-related hardware and functions.

2. Select the **Group >Targets**, **Switch > Targets**, or **Engine > Targets**, from the tool bar menu, or right-click on the group, switch, or encryption engine in the **Encryption Devices** table, and select **Targets**.

The **Encryption Targets** dialog box ([Figure 17](#)) displays the targets currently being encrypted by the selected group, switch, or encryption engine. If a group is selected, all configured targets in the group are displayed. If a switch is selected, all configured targets for the switch are displayed.

The **Encryption Targets** dialog box enables you to launch a variety of wizards and other related dialog boxes, which are defined in [Table 4](#).

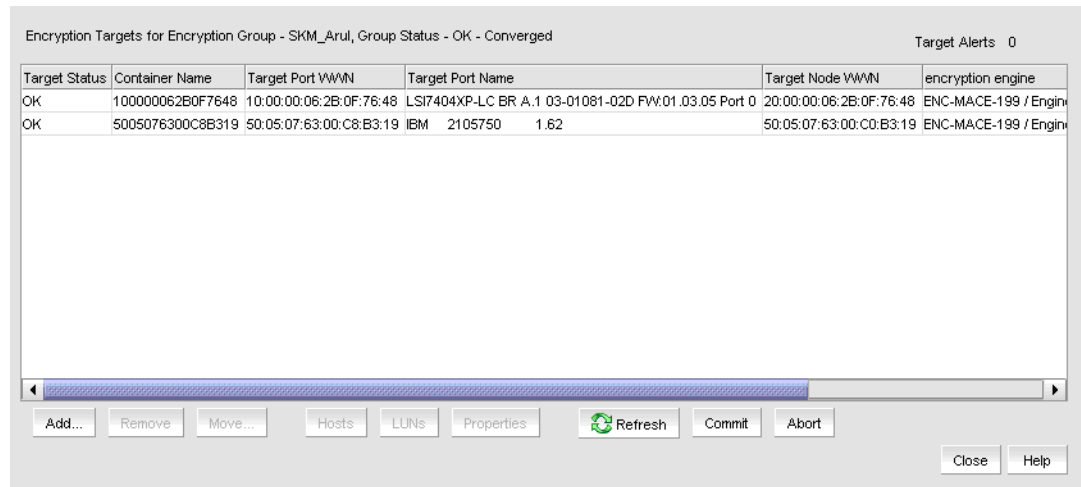


FIGURE 17 Encryption Targets dialog box

TABLE 4 Encryption Targets dialog box functionality

Feature	Description
Add button	<p>Launches the Storage Encryption Setup Wizard, which enables you to configure a new target for encryption. It is the first step in configuring encryption for a storage device.</p> <p>It is recommended that you zone the host and target together before you add container information.</p> <ul style="list-style-type: none"> Note: If the group is in OK-Converged mode, the group leader can communicate with all members. The Configure Storage Encryption wizard dialog box launches when you click Add. <p>Note:</p> <ul style="list-style-type: none"> If a group is in the Degraded state, the following operations are not allowed: key vault changes, master key operations, enable/disable encryption engines, failback mode changes, HA Cluster creation or addition (removal is allowed), tape pool changes, and any configuration changes for storage targets, hosts, and LUNs. If a group is in the Unknown state, the group leader is in an unmanaged fabric.
Remove button	<p>Removes a selected target. Proceed only if the data on the LUN is to be disabled or if the LUN is to be configured for encryption again on some other encryption engine. If the LUN data is to be enabled and later accessed by way of another encryption engine, you should unzone the host with the encryption engine <i>before</i> you remove the encryption target from the encryption engine. This prevents the host from accidentally writing to the encryption target during the unencrypted interim period.</p>



CAUTION

Removing a selected target can result in data loss, if the host is writing to the target as it is removed. Removing the target will result in lost access to the data, but the data remains encrypted on the target.

TABLE 4 Encryption Targets dialog box functionality (Continued)

Feature	Description
Move button	Moves one encryption target to a different encryption engine. The target and engine must be in the same encryption group.
Hosts button	Launches the Encryption Target Hosts dialog box, where you can configure hosts to access the selected encryption target.
LUNs button	Launches the Encryption Target LUNs dialog box, where you can display existing LUNs and add new LUNs. The button is enabled only if there are hosts associated with the targets.
Commit button	Commits LUN changes, including adding, removing, or modifying disk or tape LUNs. If there are multiple paths to the same physical LUNs, then the LUNs are added to multiple target containers (one target per storage device port). When adding, modifying, or removing multi-pathed LUNs, make the same changes in all target containers, and then click Commit to apply all the changes at once. This keeps the LUN settings consistent on each path. There is a limit of 25 LUN changes, including adding, modifying, or removing LUNs, per Commit operation. Note: The Commit button can also be used to re-create any redirection zones that were accidentally modified or removed.
Abort button	Aborts all transactions that have been configured but are not yet committed.
Properties button	Launches the Encryption Target Properties dialog box.
Refresh button	Refreshes the displayed data from the database maintained on the server. It does not collect new information from the hardware switches.

Redirection zones

It is recommended that you zone the host and target together before configuring them for encryption. Configuring a host/target pair for encryption normally creates a re-direction zone to redirect the host-target traffic through the encryption engine. But redirection zones can only be created if the host and target are already zoned. If the host and target are not already zoned, you can still configure them for encryption, but afterward you will need to zone the host and target together, and then click the **Commit** button to create the re-direction zones as a separate step.

NOTE

If you click the **Commit** button and the encryption group is busy, you are given the option to force the commit or abort the changes. Click the **Commit** button to re-create the redirection zones.

Creating a new encryption group

The following steps describe how to start and run the encryption setup wizard, and then create a new encryption group.

NOTE

When a new encryption group is created, any existing tape pools in the switch are removed.

1. Select **Configure > Encryption** from the menu bar.

The **Encryption Center** dialog box displays.

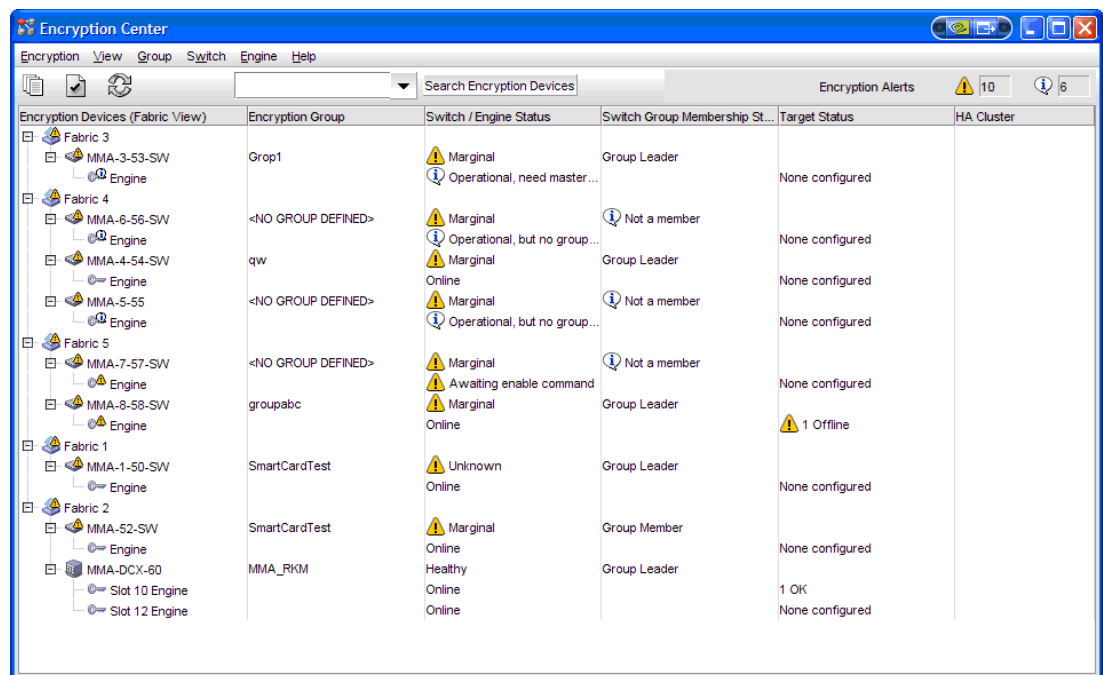


FIGURE 18 Encryption Center - No Group Defined dialog box

2. Select a switch from the **<NO GROUP DEFINED>** encryption group. The switch must not be in an encryption group already.
3. Select **Switch > Create/Add to Group**, or right-click the switch and select **Create/Add to Group**.

The **Configure Switch Encryption** welcome panel displays.

2 Creating a new encryption group

4. Click **Next**.

Create a new encryption Group is pre-selected. This is the correct selection for creating a new group.

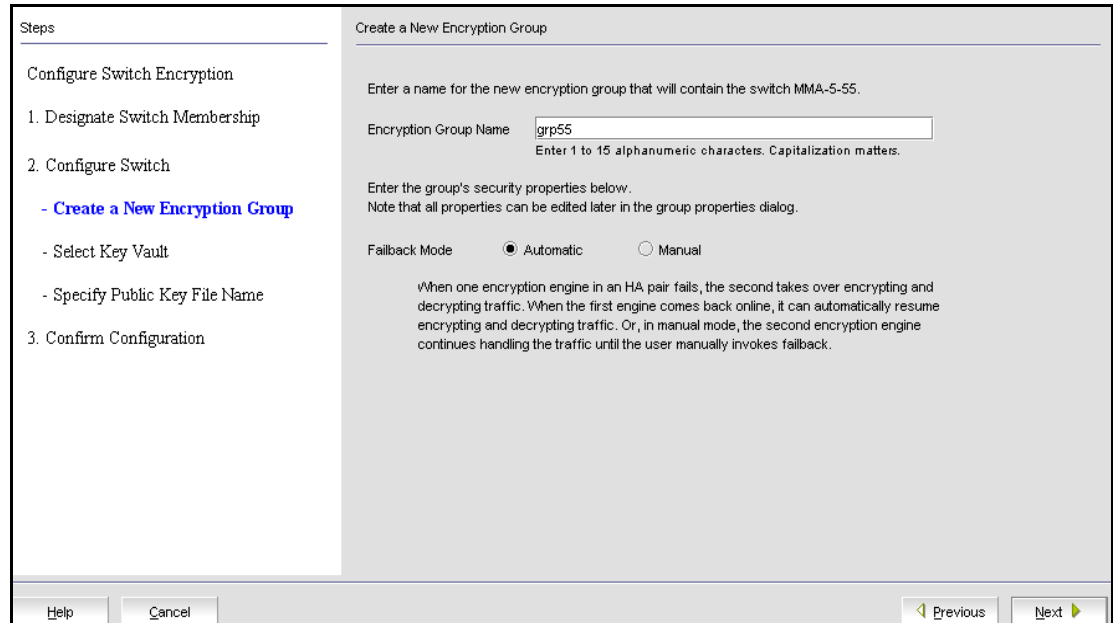


FIGURE 19 Designate Switch Membership dialog box

5. Click **Next**.

The **Create a New Encryption Group** dialog box displays.

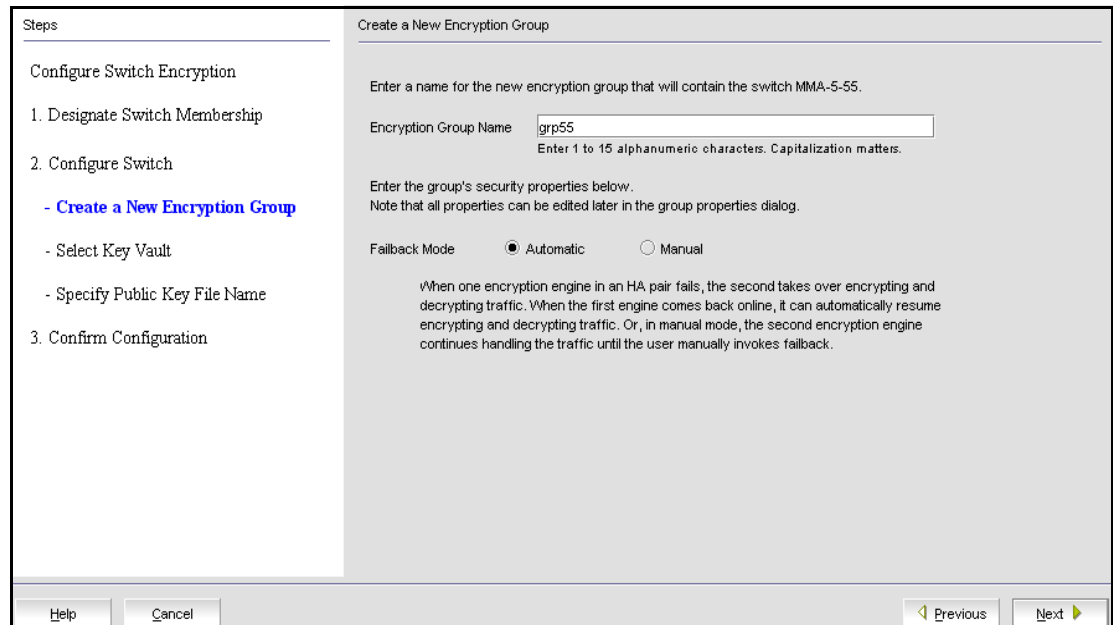


FIGURE 20 Create a new encryption group dialog box

6. Enter an **Encryption Group Name** for the encryption group (the maximum length of the group name is 15 characters; letters, digits, and underscores are allowed) and select the **Automatic** fallback mode.

NOTE

If the name you enter for the encryption group already exists, a pop-up warning message displays. Although unique group names avoid confusion while managing multiple groups, you are not prevented from using duplicate group names. Click **Yes** to use the same name for the new encryption group, or click **No** to enter a new, unique name.

7. Click **Next**.

The **Select Key Vault** dialog box displays.

FIGURE 21 Select Key Vault dialog box

8. Select the **Key Vault Type**. The choices are the following:
 - RKM - RSA Key Manager
 - LKM - NetApp Link Key Manager
 - SKM - HP Secure Key Manager
 - NCKA - Thales Encryption Manager for Storage
9. Enter the IP address or host name for the primary key vault.

When a new key vault IP address or host name is entered, you must also enter the name of the file that holds the primary key vault's public key certificate (or browse to the location by clicking the **Browse** button).
10. Enter the name of the file holding the primary key vault's public key certificate.

If you are using a backup key vault, also enter the IP address or host name, and the name of the file holding the backup key vault's public key certificate in the fields provided.

Key vault address changes

Before you add or change a key vault address, you must install the public key certificates for all switches in the encryption group on the key vault. Use the **Encryption Group Properties** dialog box to check a switch's connection status to the new key vault and to obtain the switch's public key certificate.

If you remove a primary key vault IP address, and a backup key vault has been configured, you can use the backup, but no new disk LUNs can be encrypted, no disk LUNs can be re-keyed, and no new tape LUNs can be encrypted. New tapes in a tape pool that has an existing DEK can be encrypted. Existing disk and tape LUNs can still be decrypted.

11. Click **Next**.

The **Specify Public Key Certificate Filename** panel displays.

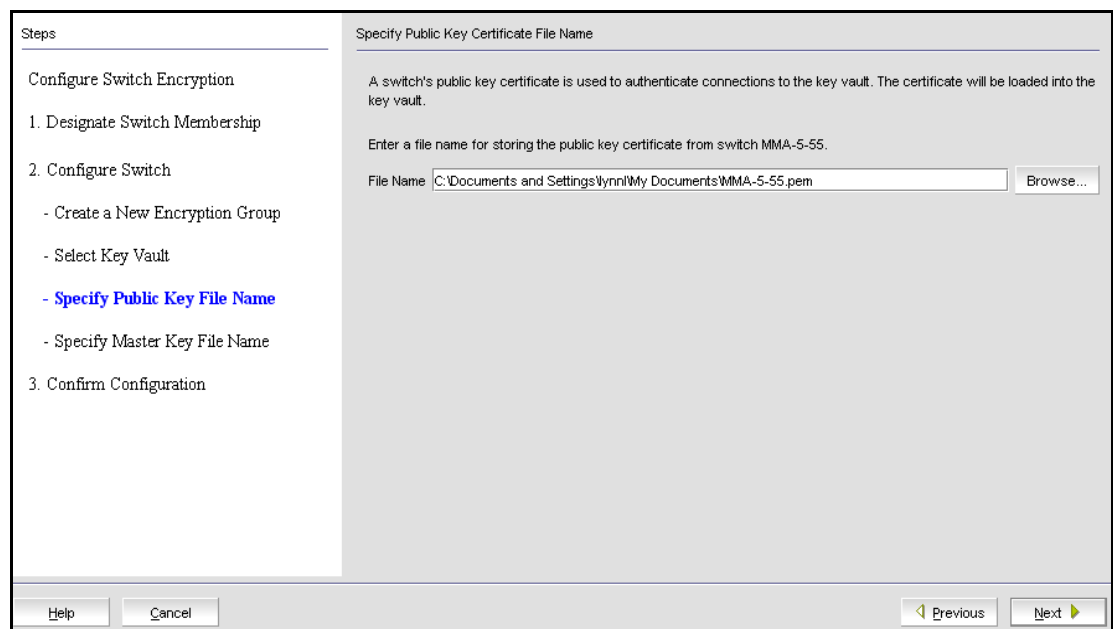


FIGURE 22 Specify Public Key Certificate filename dialog box

12. Specify the name of the file where you want to store the public key certificate that is used to authenticate connections to the key vault, and click **Next**.

The certificate stored in this file is the switch's public key certificate. You will need to know this path and file name to install the switch's public key certificate on the key management appliance.

13. Click **Next**.

If you chose LKM as the **Key Vault Type**, the **Confirm Configuration** panel displays (skip to [step 27](#)).

For all other supported key vault types, the **Specify Master Key File Name** panel displays.

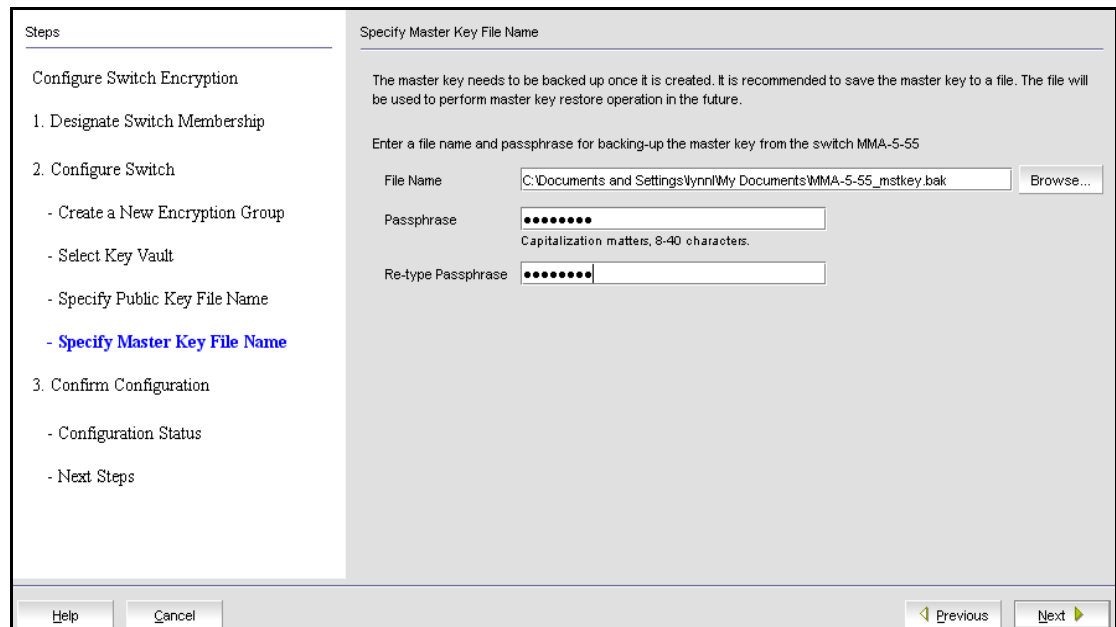


FIGURE 23 Specify Master Key File Name dialog box

14. Enter a file name, or browse to the desired location.
15. Enter the passphrase, which is required for restoring the master key. The passphrase can be between eight and 40 characters, and any character is allowed.
16. Re-type the passphrase for verification.
17. Click **Next**.

2 Creating a new encryption group

The **Select Security Settings** dialog box displays (Figure 24).

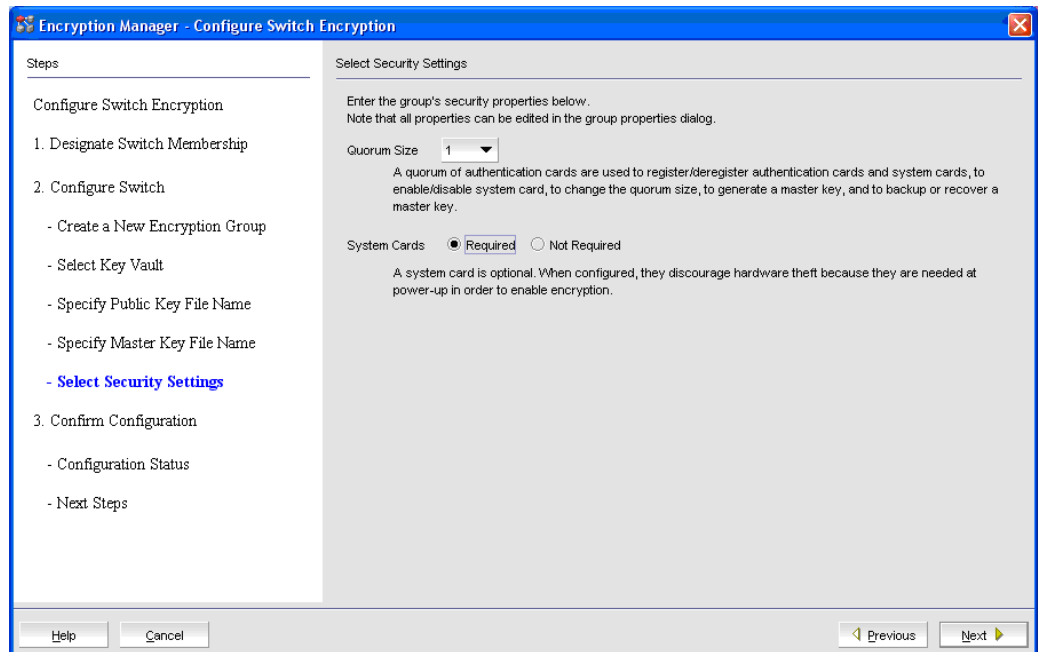


FIGURE 24 Select Security Settings dialog box

18. If you are using smart cards for authentication, specify a quorum size. The quorum size is the minimum number of cards necessary to enable the card holders to perform the security sensitive operations listed above. The maximum quorum size is five cards. The actual number of authentication cards registered is always more than the quorum size, so if you set the quorum size to five, for example, you will need to register at least six cards in the subsequent steps.
19. Set **System Cards** to **Required** to require the use a system card to control activation of an encryption engine. If **System Cards** is set to **Not Required**, the encryption engine activates without the need to read a system card first.
20. Click **Next**.

If you set a quorum size for authentication cards, the **Authentication Cards** dialog box displays (Figure 25). If you did not, the **Confirm Configuration** dialog box displays (Figure 26).

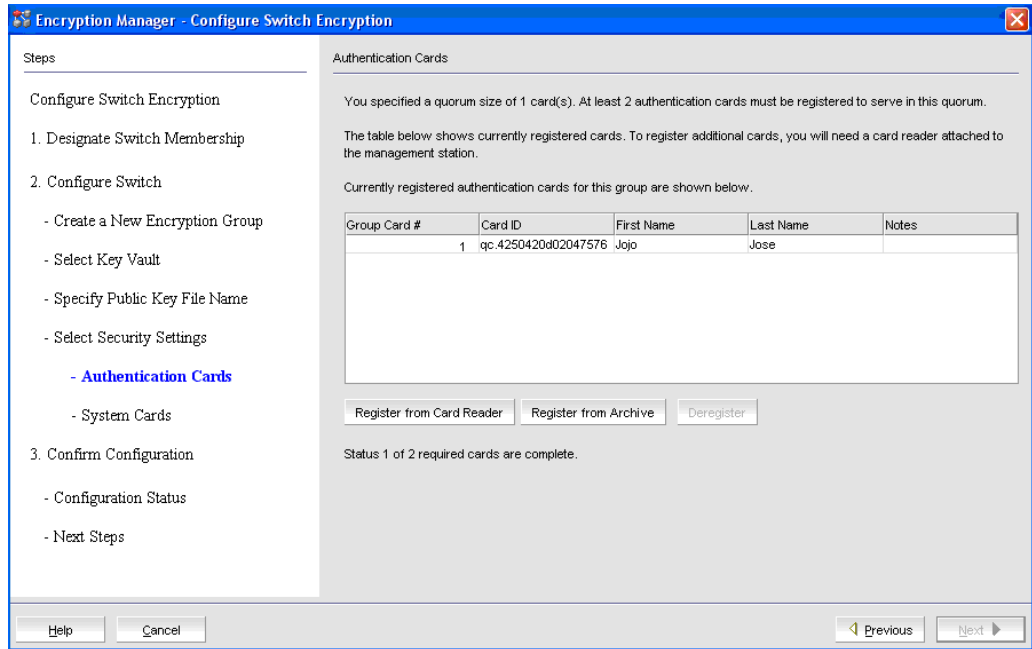


FIGURE 25 Authentication Cards dialog box

21. Select **Register from Card Reader** to register a new card.

The **Add Authentication Card** dialog box is displayed.

22. Insert a smart card into the card reader. Be sure to wait for the card serial number to appear, and then enter card assignment information, as directed.

23. Click **OK**.

24. Wait for the confirmation dialog box indicating initialization is done, and click **OK**.

The card is added to the **Registered Authentication Cards** table on the **Authentication Cards** dialog box.

25. Repeat steps 7 through 10 until you have registered all the cards, and they all display in the **Registered Authentication Cards** table on the **Authentication Cards** dialog box. Remember that you need to register the number selected as the quorum size plus at least one more card.

26. Click **Next**.

2 Creating a new encryption group

The **Confirm Configuration** panel displays the encryption group name and switch public key certificate file name you specified, shown in [Figure 26](#).

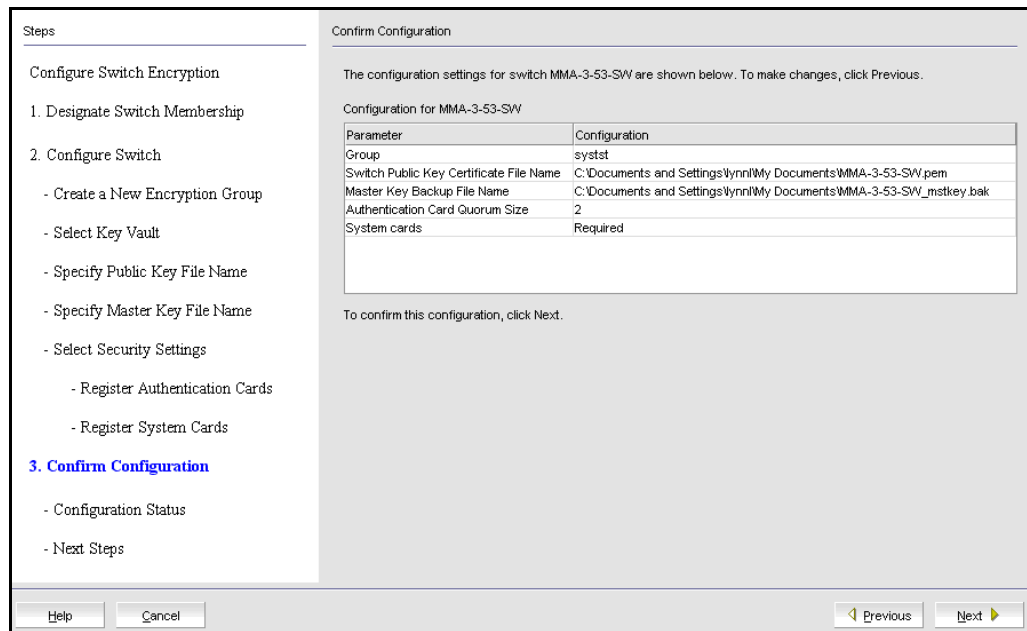


FIGURE 26 Confirm Configuration dialog box

27. Click **Next** to confirm the displayed information.

The **Configuration Status** displays, as shown in [Figure 27](#). The configuration status steps vary slightly depending on the key vault type.

- A progress indicator shows that a configuration step is in progress. A green check mark indicates successful completion of all steps for that **Configuration Item**. A red stop sign indicates a failed step.
- All **Configuration Items** have green check marks if the configuration is successful. A message displays below the table, indicating that the encryption switch was added to the group you named, and the public key certificate is stored in the location you specified.

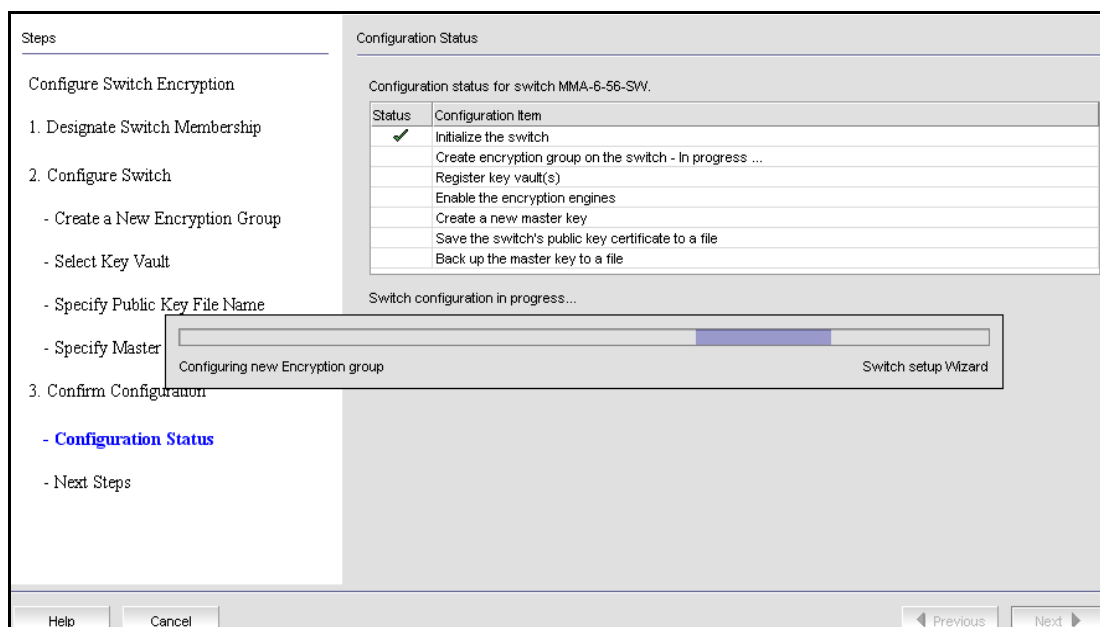


FIGURE 27 Configuration Status dialog box

The Management application sends API commands to verify the switch configuration. The CLI commands are detailed in the *Fabric OS Encryption Administrator's Guide*, "Key vault configuration."

- **Initialize the switch**
If the switch is not already in the initiated state, the Management application performs the `cryptocfg --initnode` command.
- **Create encryption group on the switch**
The Management application creates a new group using the `cryptocfg --create -encgroup` command, and sets the key vault type using the `cryptocfg --set -keyvault` command.
- **Register key vault(s)**
The Management application registers the key vault using the `cryptocfg --reg keyvault` command.
- **Enable the encryption engines**
The Management application initializes an encryption switch using the `cryptocfg --initEE [<slotnumber>]` and `cryptocfg --regEE [<slotnumber>]` commands.

2 Creating a new encryption group

- **Create a new master key**

The Management application checks for a new master key. New master keys are generated from the Encryption Group Properties dialog box, Security tab. See “[Creating a new master key](#)” on page 76 for more information.

- **Save the switch’s public key certificate to a file**

The Management application saves the KAC certificate into the specified file.

- **Back up the master key to a file**

The Management application saves the master key into the specified file. Note that a master key is not generated if the key vault type is LKM. LKM manages DEK exchanges through a trusted link, and the LKM appliance uses its own master key to encrypt DEKs.

28. Click **Next**.

The **Read Instructions** dialog box displays instructions for installing public key certificates for the encryption switch. These instructions are specific to the key vault type. Copy or print these instructions.

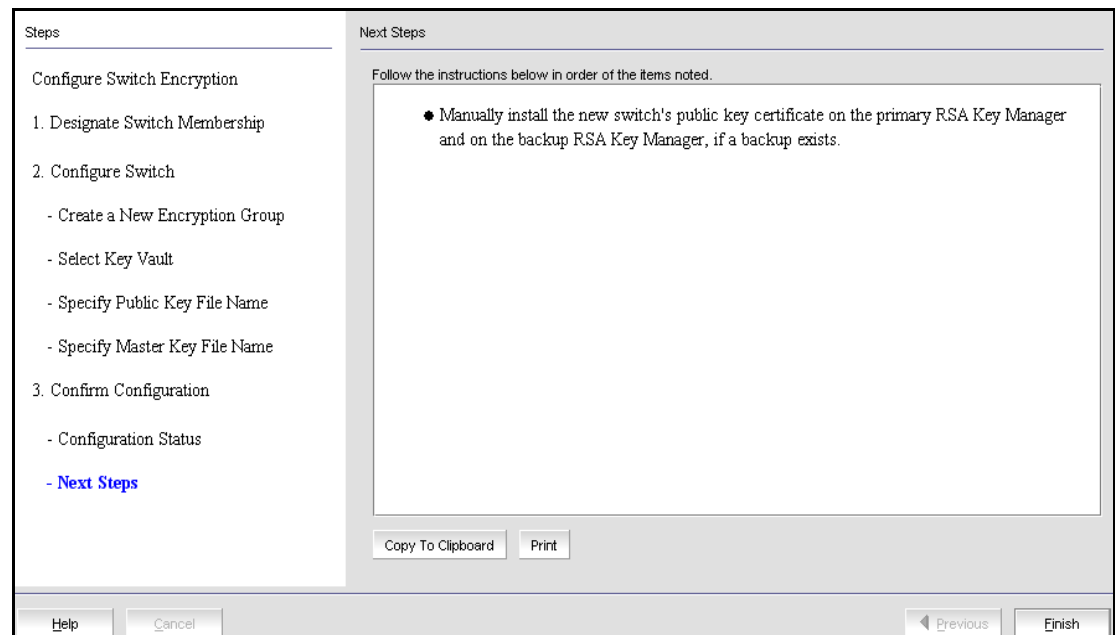


FIGURE 28 Read Instructions dialog box

29. Click **Finish** to exit the **Configure Switch Encryption** wizard.

At this point, a **Next Steps** dialog box is displayed, with brief instructions that are specific to certificate exchanges between the switch and key manager you are using. Refer to [Appendix D, “Supported Key Management Systems”](#) for more detailed instructions for certificate exchange with each supported key manager, and refer to the key manager user documentation for additional information.

Adding a switch to an encryption group

The setup wizard allows you to either create a new encryption group, or add an encryption switch to an existing encryption group. Use the following procedure to add a switch to an encryption group.

1. Select **Configure > Encryption** from the menu bar.
The **Encryption Center** dialog box displays.
2. Select the switch to be added to the group. The switch must not already be in an encryption group.
3. Select **Switch > Create/Add to Group**, or right-click the switch and select **Create/Add to Group**.
The **Configure Switch Encryption** welcome panel displays.
4. Click **Next**.
The **Designate Switch Membership** panel displays.

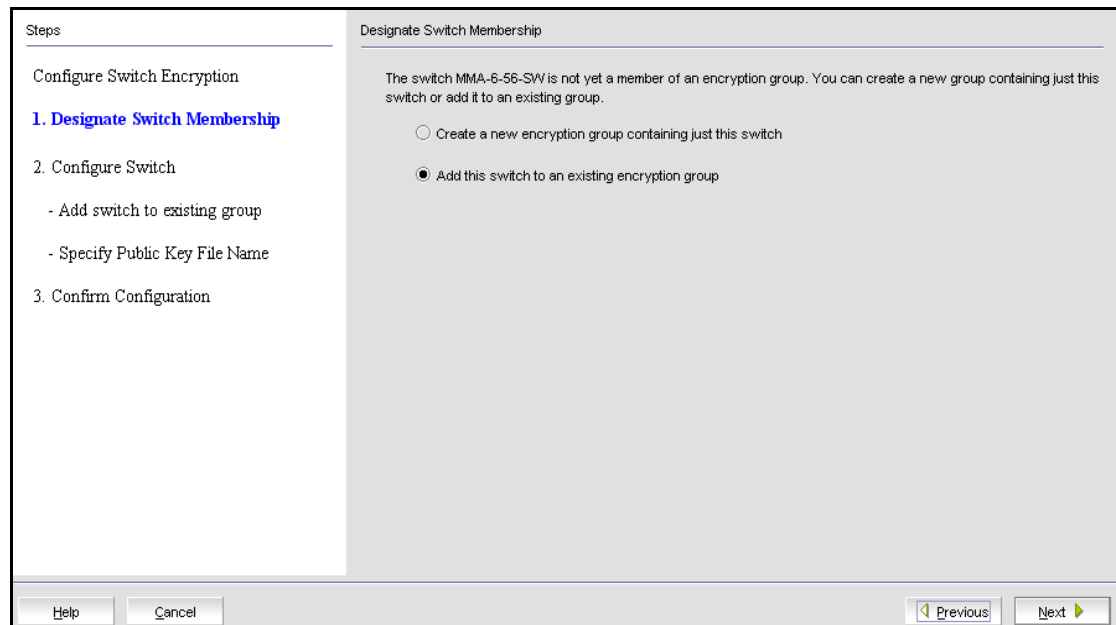


FIGURE 29 Add switch to an encryption group - Designate Switch Membership dialog box

- a. Select **Add this switch to an existing encryption group**.
- b. Click **Next**.

The **Add Switch to Existing Encryption Group** dialog box displays.

2 Adding a switch to an encryption group

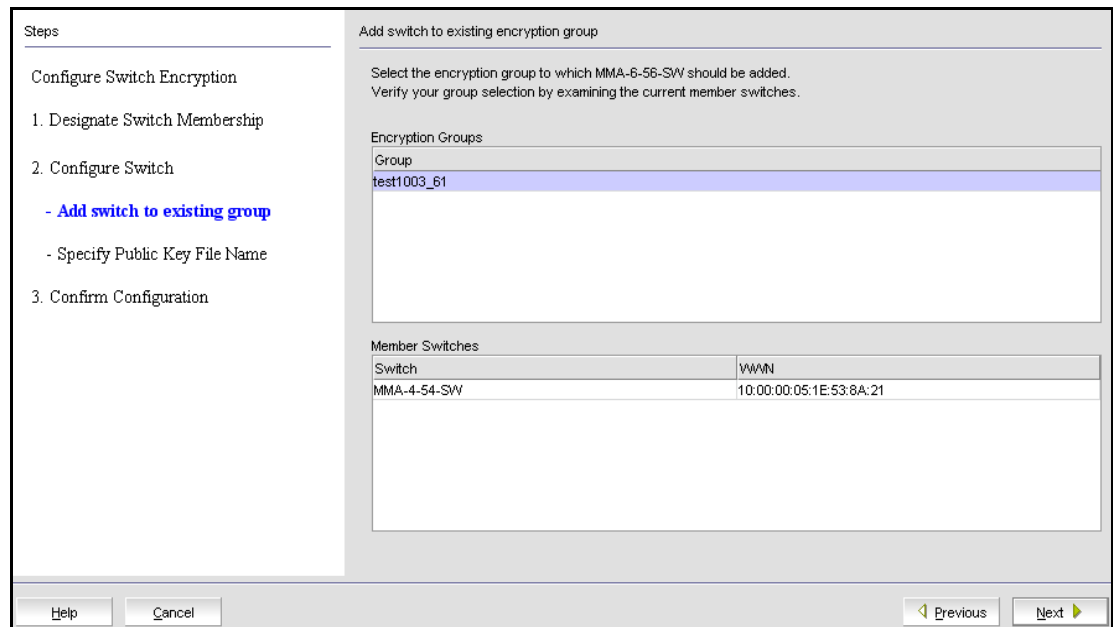


FIGURE 30 Add Switch to Existing Encryption Group dialog box

5. Select the group to which you want to add the switch, and click **Next**.
The **Specify Public Key Certificate Filename** panel displays.

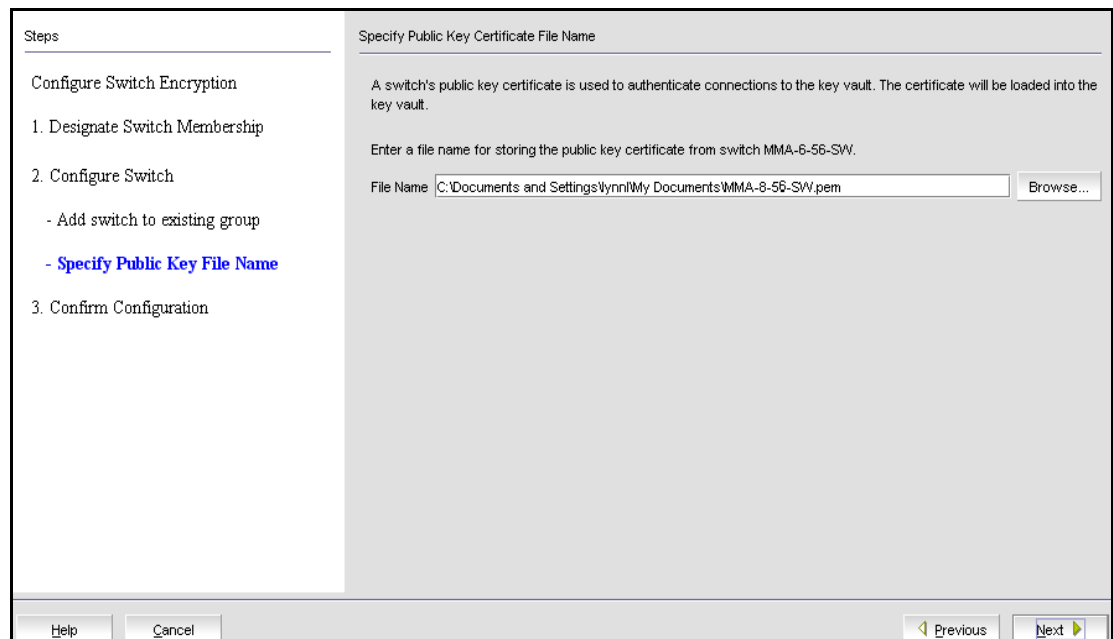


FIGURE 31 Add switch to an encryption group - Specify Public Key Certificate filename dialog box

6. Specify the name of the file where you want to store the public key certificate that is used to authenticate connections to the key vault, and click **Next**.

The **Confirm Configuration** panel displays the encryption group name and switch public key certificate file name you specified.

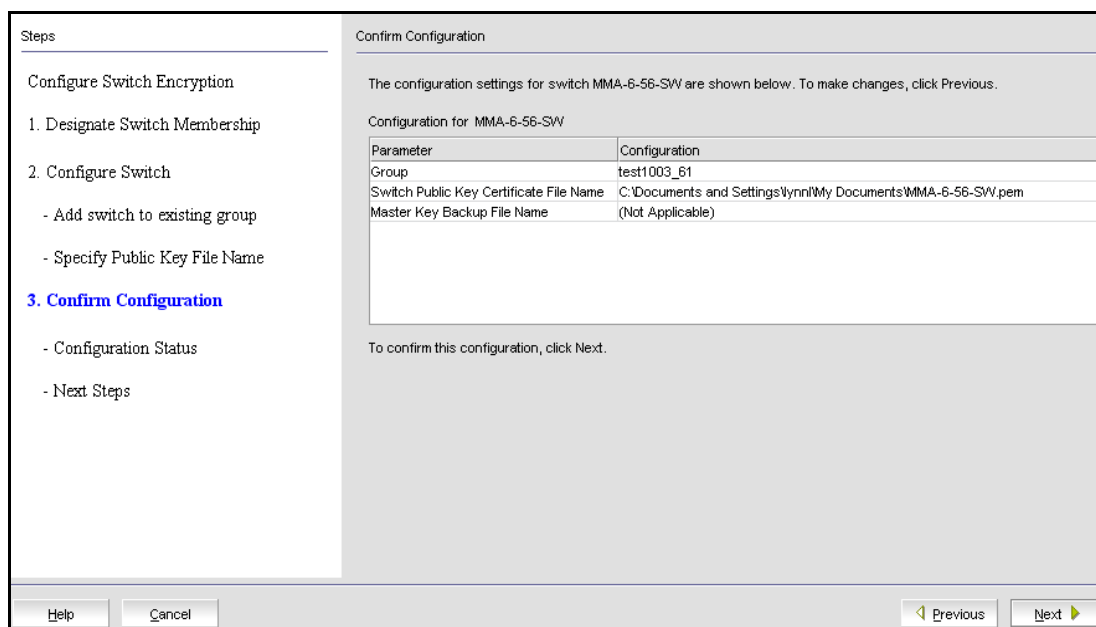


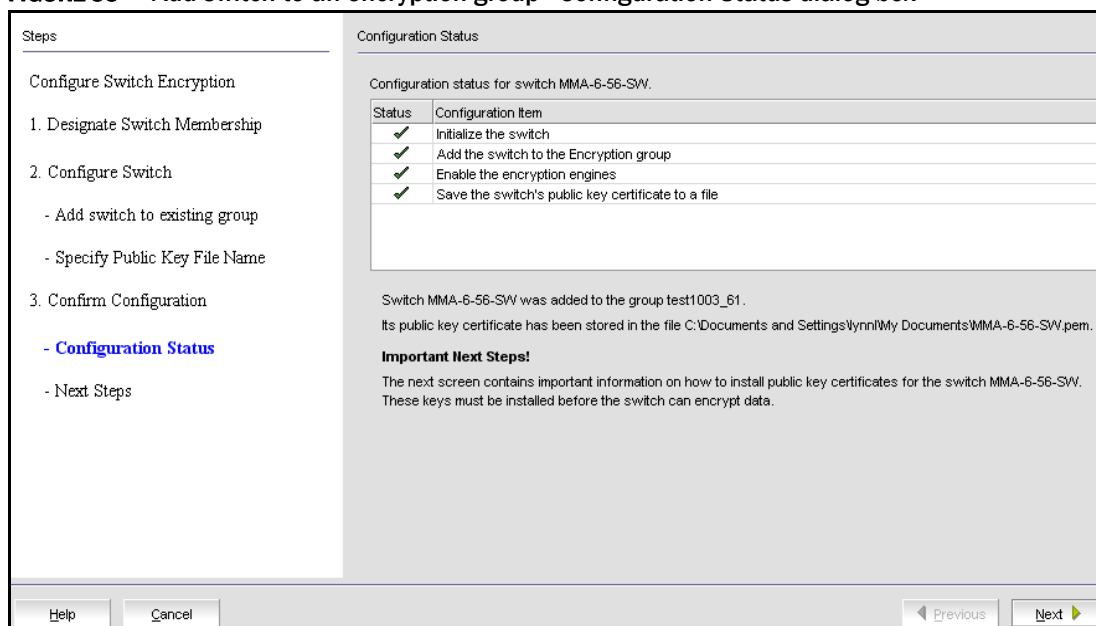
FIGURE 32 Add switch to an encryption group - Confirm Configuration dialog box

7. Click **Next** to confirm the displayed information.

The **Configuration Status** displays.

- A progress indicator shows that a configuration step is in progress. A green check mark indicates successful completion of all steps for that **Configuration Item**. A red stop sign indicates a failed step.
- All **Configuration Items** have green check marks if the configuration is successful. A message displays below the table, indicating that the encryption switch was added to the group you named, and the public key certificate is stored in the location you specified.

FIGURE 33 Add switch to an encryption group - Configuration Status dialog box



2 Creating high availability (HA) clusters

8. Note **Important Next Steps!** below this message, and click **Next**.

Instructions for installing public key certificates for the encryption switch are displayed. These instructions are specific to the key vault type. Copy or print these instructions.

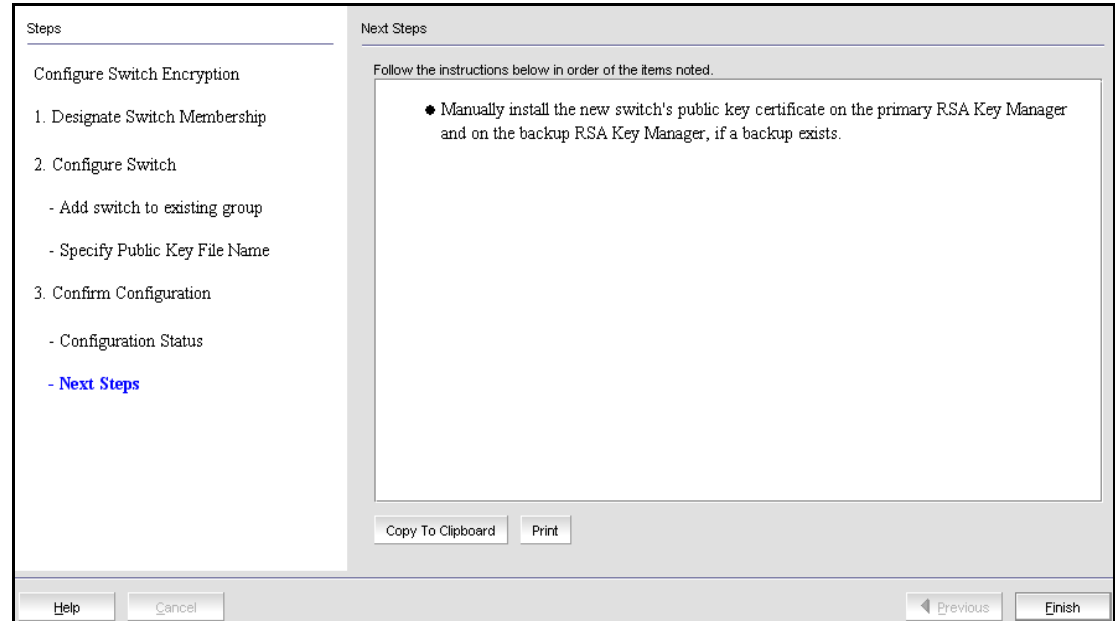


FIGURE 34 Add switch to an encryption group - Next Steps dialog box

9. Click **Finish** to exit the **Configure Switch Encryption** wizard.

Creating high availability (HA) clusters

A high availability (HA) cluster is a group of exactly two encryption engines. One encryption engine can take over encryption and decryption tasks for the other encryption engine, if that member fails or becomes unreachable.

The following rules apply when configuring an HA cluster:

- All HA cluster configuration and related operations must be performed on the group leader.
- I/O sync links must be configured before creating an HA cluster. Refer to the Brocade Fabric OS Encryption Administrator's Guide for information about I/O sync link configuration.
- Configuration changes must be committed before they take effect. Any operation related to an HA cluster that is performed without a commit operation will not survive across switch reboots, power cycles, CP failover, or HA reboots.
- It is recommended that the HA cluster configuration be completed before you configure storage devices for encryption.
- It is mandatory that the two encryption engines in the HA cluster belong to two different nodes for true redundancy. This is always the case for Brocade encryption switches, but is not true if two FS8-18 blades in the same DCX or DCX-4S chassis are configured in the same HA cluster. In Fabric OS version 6.3.0 and later releases, HA cluster creation is blocked when encryption engines belonging to FS8-18 blades in the same DCX or DCX-4S are specified.

When creating a new HA Cluster, add one engine to create the cluster and then add the second engine. You can make multiple changes to the HA Clusters list; the changes are not applied to the switch until you click **OK**.

Both engines in an HA cluster must be in the same fabric as well as the same encryption group.

1. Select **Configure > Encryption** from the menu bar.

The **Encryption Center** dialog box displays.

2. If groups are not visible in the **Encryption Devices** table, select **View > Groups** from the menu bar.

The encryption groups display in the **Encryption Devices** table.

3. Select an encryption group from the tree, and select **Group > HA Cluster** from the menu bar, or right-click the encryption group and select **HA Cluster**.

Encryption Group Properties are displayed, with the **HA Clusters** tab selected (Figure 35). Available encryption engines are listed under **Non-HA Encryption Engines**.

4. Select an available encryption engine, and a destination HA cluster under **High-Availability Clusters**. Select **New HA Cluster** if you are creating a new cluster.
5. Click the right arrow to add the encryption engine to the selected HA cluster.

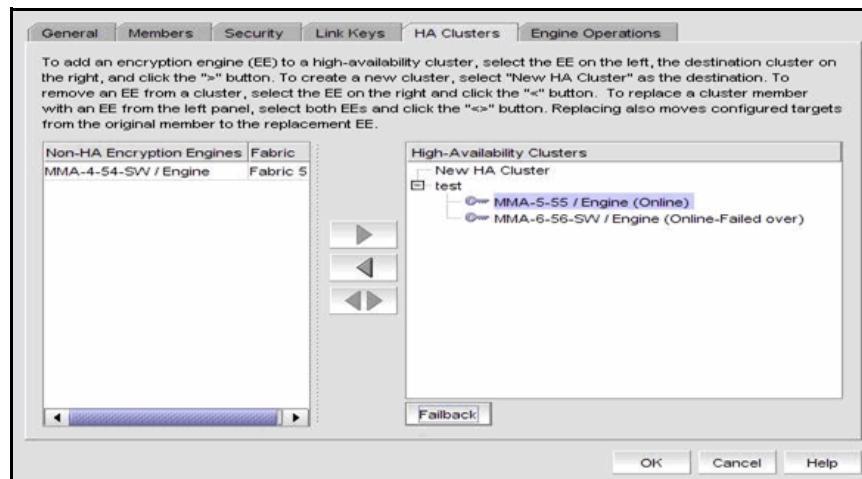


FIGURE 35 HA Clusters tab

NOTE

If you are creating a new HA cluster, a dialog box displays requesting a name for the new HA cluster. HA Cluster names can have up to 31 characters. Letters, digits, and underscores are allowed.

Removing engines from an HA cluster

Removing the last engine from an HA cluster also removes the HA cluster.

If only one engine is removed from a two-engine cluster, you must either add another engine to the cluster or the other engine must be removed too.

2 Creating high availability (HA) clusters

1. Select an encryption engine from the right tree (see [Figure 35](#)) and click the left arrow button.
2. Either remove the second engine or add a replacement second engine, making sure all HA clusters have exactly two engines.
3. Click **OK**.

Swapping engines in an HA cluster

Swapping engines is useful when replacing hardware. Swapping engines is different from removing an engine and adding another because when you swap engines, the configured targets on the former HA cluster member are moved to the new HA cluster member.

To swap engines, select one engine from the right tree (see [Figure 35](#)) and one unclustered engine from the list on the left, and click the double-arrow button.

NOTE

The two engines being swapped must be in the same fabric.

Failback option

The **Failback** option determines the behavior when a failed encryption engine is restarted. When the first encryption engine comes back online, the encryption group's failback setting (auto or manual) determines how the encryption engine resumes encrypting and decrypting traffic to its encryption targets.

- In auto mode, when the first encryption engine restarts, it automatically resumes encrypting and decrypting traffic to its encryption targets.
- In manual mode, the second encryption engine continues handling the traffic until you manually invoke failback using the CLI or Management application, or until the second encryption engine fails.

Invoking failback

To invoke failback to the restarted encryption engine from the Management application, complete the following steps.

1. Select **Configure > Encryption**.

The **Encryption Center** dialog box displays.

2. Select the group to which the encryption engine belongs from the **Encryption Devices** table, and click **Properties**.

The **Encryption Group Properties** dialog box displays.

3. Click the **HA Clusters** tab.
4. Select the online encryption engine and click **Failback**.
5. Click **OK** on the **Encryption Group Properties** dialog box.
6. Click **Close** on the **Encryption Center** dialog box.

Adding encryption targets

Adding an encryption target maps storage devices and hosts to virtual targets and virtual initiators within the encryption switch.

NOTE

It is recommended that you zone the host and target together before configuring them for encryption. If the host and target are not already zoned, you can still configure them for encryption, but afterward you will need to zone the host and target together, and then click the **Commit** button to commit the changes. If you attempt to close the Encryption Targets dialog box without committing the changes, you are reminded of uncommitted changes in the Management application.

1. Select **Configure > Encryption** from the menu bar.

The **Encryption Center** dialog box displays the status of all encryption-related hardware and functions at a glance. It is the single launching point for all encryption-related configuration

2. Select the encryption group, switch, or encryption engine to which you want to add the target.
3. Click **Encryption Targets**.

The **Encryption Targets** dialog box displays.

4. Click **Add**.

The **Configure Storage Encryption** welcome panel displays. The welcome panel explains the wizard's purpose, which is to configure encryption for a storage device (target).

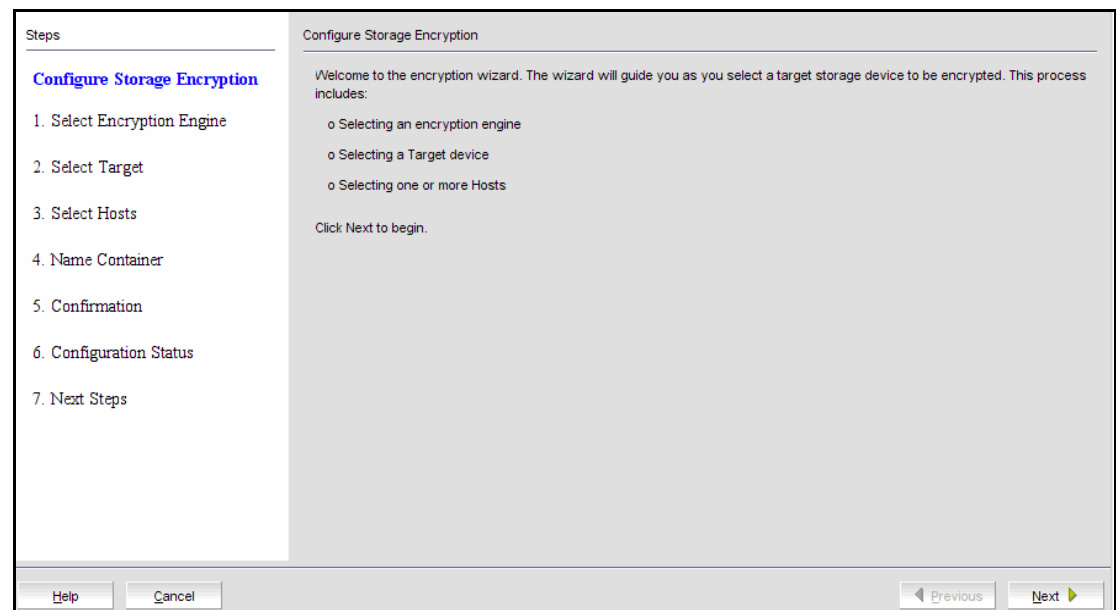


FIGURE 36 Configure Storage Encryption welcome panel

5. Click **Next** to begin.

The **Select Encryption Engine** dialog box displays. The list of engines depends on the scope being viewed.

- If the Targets dialog box is showing all targets in an encryption group, the list includes all engines in the group.
- If the Targets dialog box is showing all targets for a switch, the list includes all encryption engines for the switch.
- If the Targets dialog box is showing targets for a single encryption engine, the list contains only that engine.

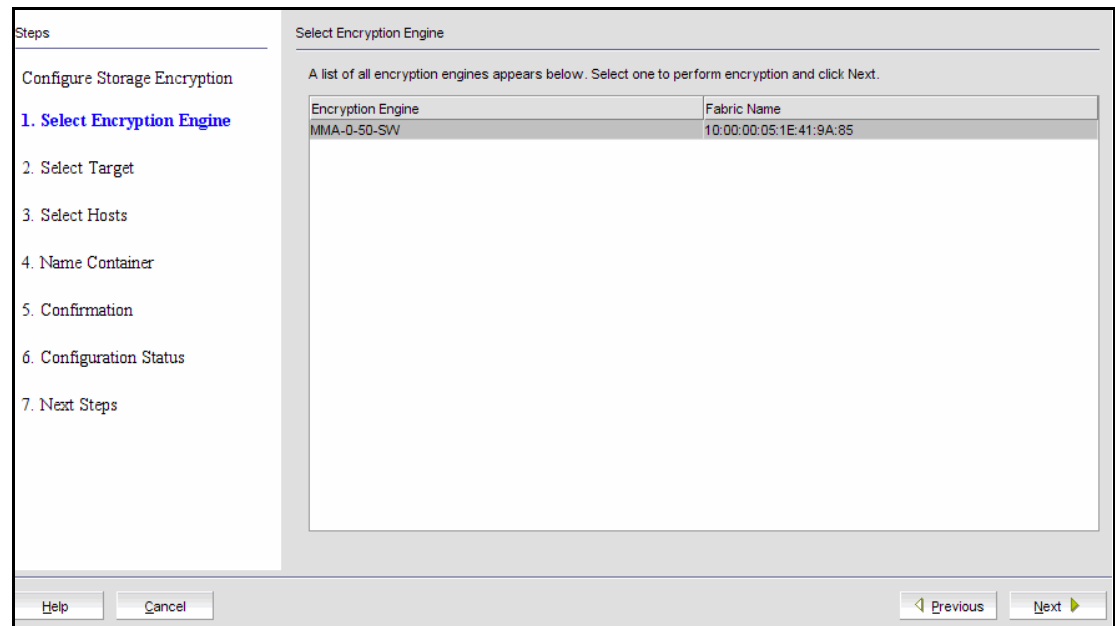


FIGURE 37 Select Encryption Engine dialog box

2 Adding encryption targets

6. Select the encryption engine (blade or switch) you want to configure, and click **Next**.

The **Select Target** panel displays. This panel lists all target ports and target nodes in the same fabric as the encryption engine. The **Select Target** list does *not* show targets that are already configured in an encryption group.

There are two available methods for selecting targets: select from the list of known targets or manually enter the port and node WWNs.

Steps

Configure Storage Encryption

1. Select Encryption Engine
- 2. Select Target**
3. Select Hosts
4. Name Container
5. Confirmation
6. Configuration Status
7. Next Steps

Select Target

You have chosen to configure a target for encryption engine 10:00:00:05:1E:41:9A:85 in switch MMA-0-50-SW in the 10:00:00:05:1E:41:9A:85 fabric. All target device port WWNs in that fabric appear below. Select a target, or enter a PORT WWN and a Node WWN in the fields below the table. Then select the target type. For information on MPIO-based configurations, click Help. When ready, click Next.

Targets in Fabric

Port WWN	Port Name	Node WWN
50:01:10:A0:00:8B:F6:0B	HP Ultrium 3 Fibre Channel S/N-HU10739...	50:01:10:A0:00:8B:F6:09
50:01:10:A0:00:8B:F6:0A	HP Ultrium 3 Fibre Channel S/N-HU10739...	50:01:10:A0:00:8B:F6:09
21:00:00:20:37:F2:C2:C4	SEAGATE ST318451FC F26D	20:00:00:20:37:F2:C2:C4
21:00:00:20:37:EC:72:0C	SEAGATE ST318451FC F28D	20:00:00:20:37:EC:72:0C
21:00:00:20:37:F2:BE:12	SEAGATE ST318451FC F26D	20:00:00:20:37:F2:BE:12
21:00:00:20:37:E8:47:31	SEAGATE ST318451FC F28D	20:00:00:20:37:E8:47:31
21:00:00:20:37:F5:B3:9A	SEAGATE ST318451FC F28D	20:00:00:20:37:F5:B3:9A
21:00:00:20:37:F2:BD:F8	SEAGATE ST318451FC F26D	20:00:00:20:37:F2:BD:F8
20:04:00:A0:B8:1F:C5:0F	LSI INF-01-00 0612	20:04:00:A0:B8:1F:C5:0D
20:04:00:A0:B8:1F:C5:0E	LSI INF-01-00 0612	20:04:00:A0:B8:1F:C5:0D
50:06:01:62:10:60:06:3A	DGC LUNZ 0219	50:06:01:60:90:60:06:3A
10:00:00:06:2B:0C:C8:54	LSI7202XP-LC A.1 03-00021-02B FW:01....	20:00:00:06:2B:0C:C8:54
10:00:00:06:2B:0D:30:50	LSI7204XP-LC A.1 03-01078-02F FW:01....	20:00:00:06:2B:0D:30:50

Port WWN

Node WWN

Type

Help Cancel Previous Next

FIGURE 38 Select Target dialog box

- a. Select a target from the list. (The **Target Port WWN** and **Target Node WWN** fields contain all the target information that displays using the `nsshow` command.) You can also enter WWNs manually if you prefer, or if you want to specify a target that is not on the list.
- b. Select a **Target Type**. Disk is selected and cannot be changed. If the target node is disk storage, choose **Disk**. If the target port is tape storage, choose **Tape**.

7. Click **Next**.

The **Select Hosts** panel displays. This panel lists all hosts in the same fabric as the encryption engine. There are two available methods for selecting hosts: select from a list of known hosts or manually enter the port and node world wide names.

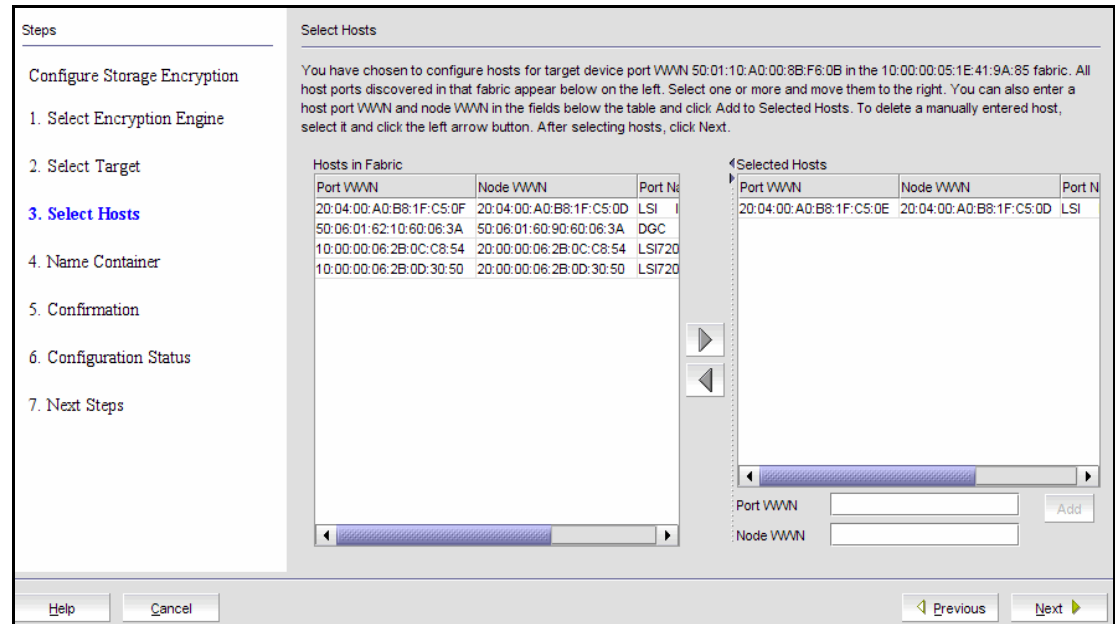


FIGURE 39 Select Hosts dialog box

- a. Select a maximum of 1024 hosts from the **Host Ports in Fabric** list, and click the right arrow to move the host to the **Selected Hosts** list. (The **Host Port WWN** column contains all the target information that displays using the `nsshow` command.)
 - b. Manually enter world wide names in the **Host Port WWN** and **Host Node WWN** text boxes, if the hosts are not included in the list. You must fill in both the Host Port WWN and the Host Node WWN. Click the **Add to Selected Hosts** button to move the host to the **Selected Hosts** list.
8. Click **Next** when you are finished selecting hosts or manually entering the WWNs.

The **Name Container** panel displays.

The name container step in the wizard enables you to specify a name for the target container that is created in the encryption engine to hold the target configuration data.

9. The container name defaults to the target WWPN. You can, however, rename the container name. If you want to specify a name other than the default, enter a name, using a maximum number of 31 characters. Letters, digits, and underscores are allowed.

2 Adding encryption targets

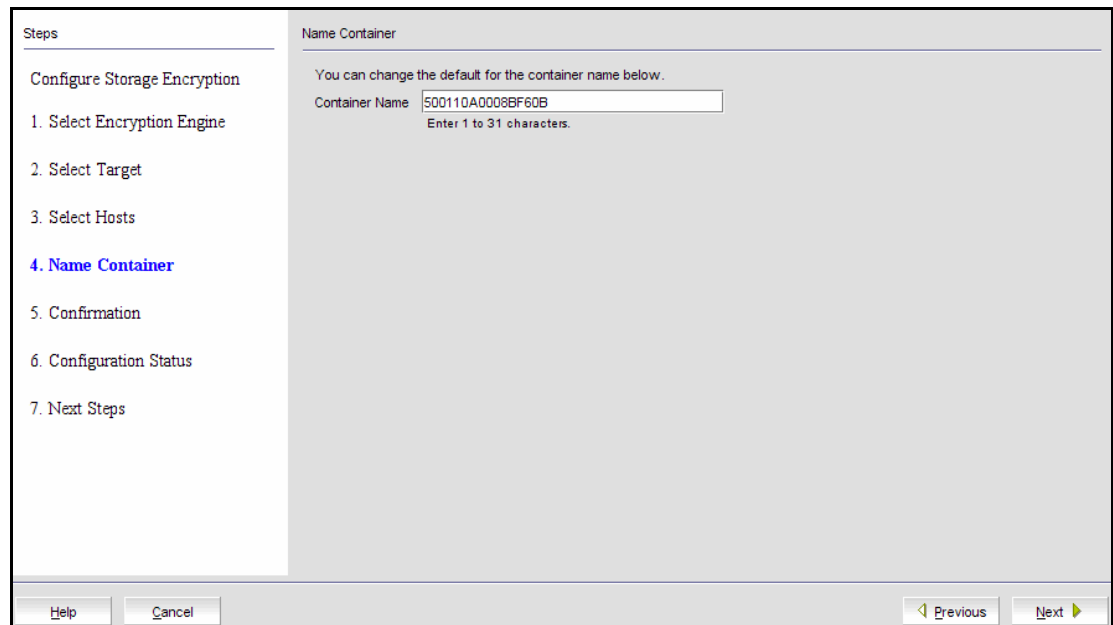


FIGURE 40 Name Container dialog box

10. Click **Next**.

The **Confirmation** panel displays.

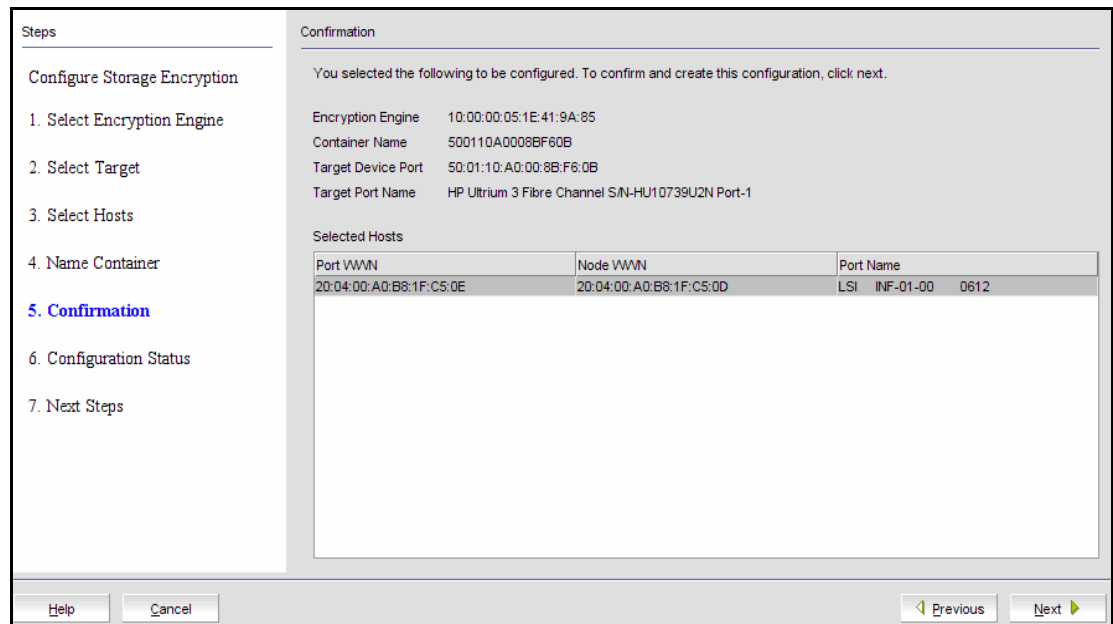


FIGURE 41 Confirmation dialog box

11. Click **Next** to confirm the displayed information.

The **Configuration Status** displays the target and host that are configured in the target container, as well as the virtual targets (VT) and virtual initiators (VI).

NOTE

If you can view the VI/VT Port WWNs and VI/VT Node WWNs, the container has been successfully added to the switch.

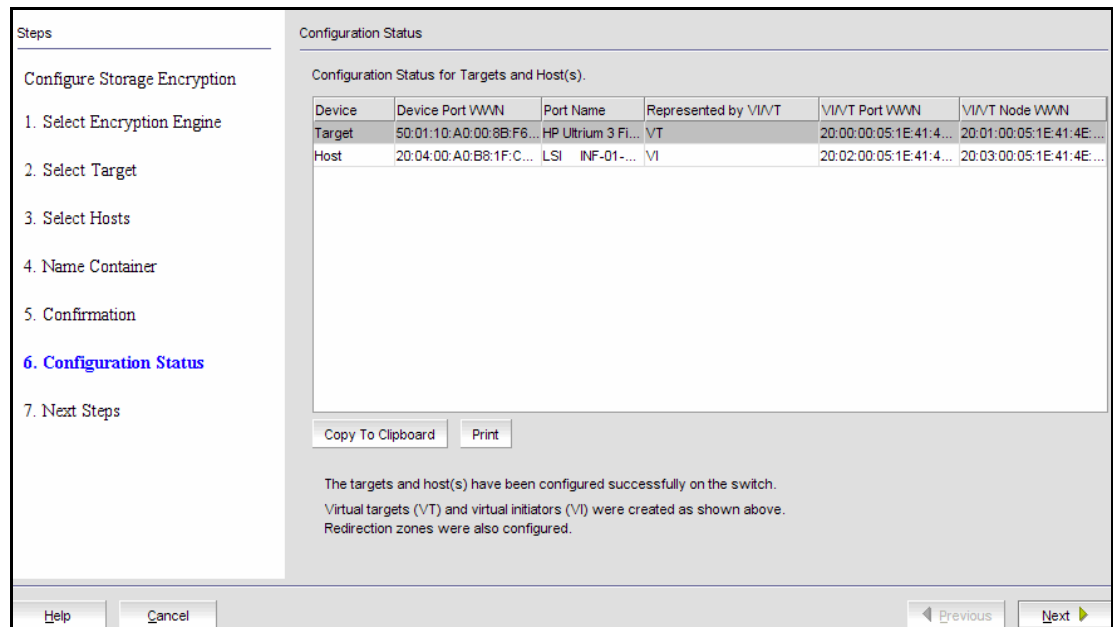


FIGURE 42 Configuration Status dialog box

12. Review the configuration. If you want to save a copy of the instructions, click the **Copy to Clipboard** button.

2 Adding encryption targets

13. Click **Next** to confirm the configuration.

The **Important Instructions** dialog box displays.

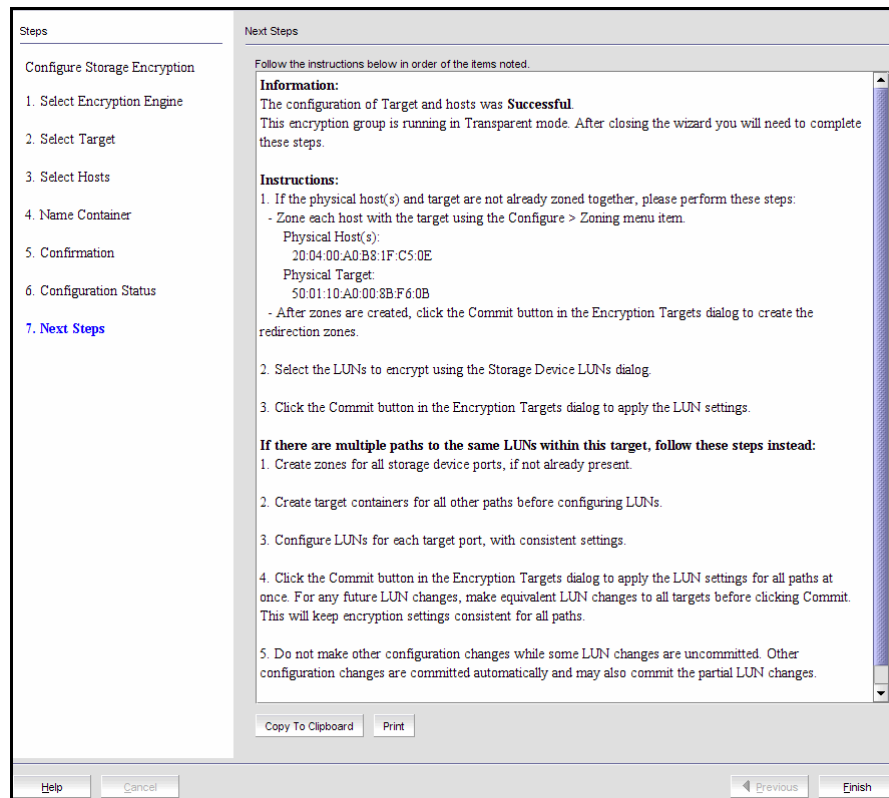


FIGURE 43 Important Instructions dialog box

14. Review the instructions about post-configuration tasks you must complete after you close the wizard.

15. Click **Finish** to exit the **Configure Storage Encryption** wizard.

Configuring hosts for encryption targets

Use the **Encryption Target Hosts** dialog box to edit (add or remove) hosts for an encrypted target.

NOTE

Hosts are normally selected as part of the **Configure Storage Encryption** wizard but you can also edit hosts later using the **Encryption Target Hosts** dialog box.

1. Select **Configure > Encryption** from the menu bar.
The **Encryption Center** dialog box displays.
2. Select the encryption group, switch, or encryption engine containing the storage device to be configured.
3. Click **Encryption Targets**.
The **Encryption Targets** dialog box displays.
4. Select a Target storage device from the list, and click **Hosts**.
The **Encryption Target Hosts** dialog box displays. This dialog box lists configured hosts in a fabric.
The **Encryption Target Hosts** dialog box displays. This dialog box lists configured hosts in a fabric.
5. Select one or more hosts in a fabric and move them to the **Selected Hosts** table.

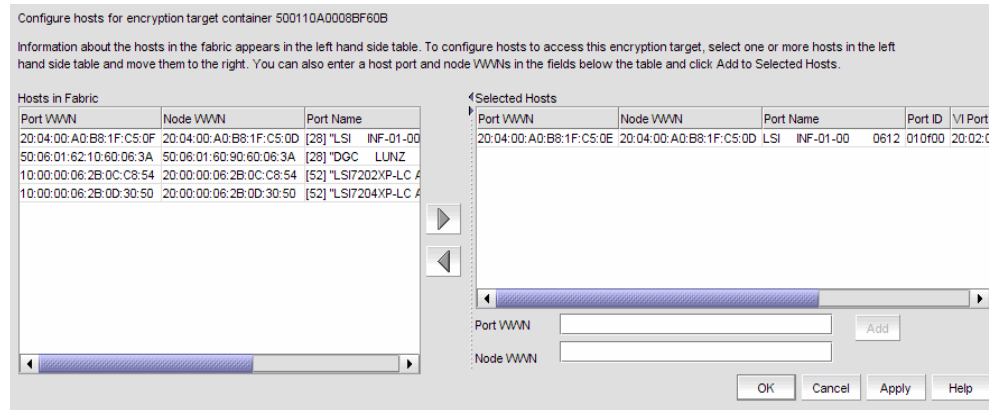


FIGURE 44 Encryption Target Hosts dialog box

Adding Target Disk LUNs for encryption

The **Encryption Target LUNs** dialog box lists configured LUNs. The displayed information is different for disk and tape devices. For example, tape volume and label information is included for tape devices. Initially, this list is empty.

NOTE

If you are using VMware virtualization software or any other configuration that involves mounted file systems on the LUN, you must enable first-time encryption when you create the LUN.

You configure a Crypto LUN by adding the LUN to the CryptoTarget container and enabling the encryption property on the Crypto LUN. You must add LUNs manually. The LUNs of the target which are not enabled for encryption must still be added to the CryptoTarget container with the **Clear Text** encryption mode option.

NOTE

When configuring a LUN with multiple paths, the same LUN policies must be configured on all the LUN's paths. If there are multiple paths to the same physical LUNs, then the LUNs are added to multiple target containers (one target per storage device port). See [“Configuring encrypted storage in a multi-path environment”](#) on page 66 for a multi-path configuration scenario.

-
1. Select **Configure > Encryption** from the menu bar.

The **Encryption Center** dialog box displays.

2. Select the encryption group, switch, or encryption engine containing the storage device to be configured.

3. Click **Encryption Targets**.

The **Encryption Targets** dialog box displays.

4. Select a Target storage device from the list, and click **LUNs**.

The **Encryption Target LUNs** dialog box displays. Initially, this list is empty. You must add LUNs manually.

- Click the **Copy Settings** button to copy the data from a selected row to the next row.
- Click the **Re-keying Details** button to launch the **LUN Re-keying Details** dialog of the selected LUN. When re-keying is in progress, the re-key completion percentage is updated automatically, at one minute intervals, until completion.

NOTE

You must configure LUNs on storage devices that are listed in the **Targets** dialog box for the host to access them, even if the LUNs are not encrypted.

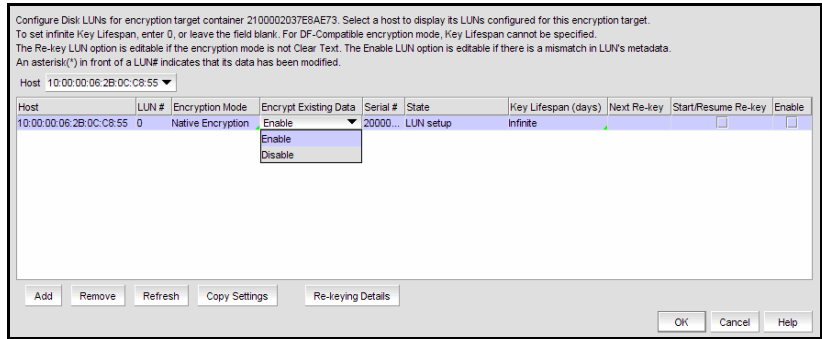


FIGURE 45 Encryption Target Disk LUNs dialog box

5. Click **Add**.

The **Add LUNs** dialog box displays.

This dialog box includes a table of all LUNs in the storage device that are visible to hosts. LUNs are identified by serial number, or by host WWN and LUN number. The LUN numbers may be different for different hosts.

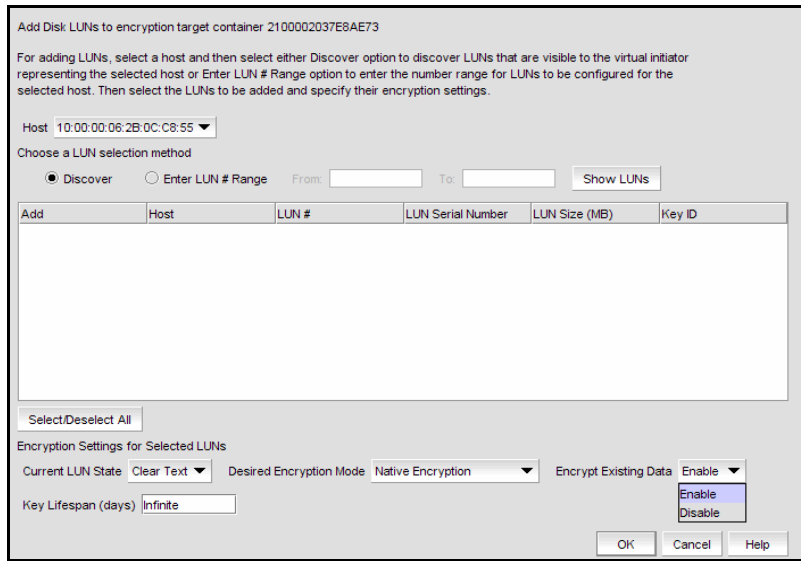


FIGURE 46 Add Encryption Target Disk LUNs dialog box

6. Select a host from the **Host** list.

There are two possible sources for the list of LUNs:

- Specify a range of LUN numbers and click **Show LUNs**. This fills the table with dummy LUN information. This method works even if the target is offline. You can specify a range of LUN numbers only if a host is chosen from the list. If **All Hosts** is selected, you will not be able to specify a range but can discover LUNs.
- Request discovery and click **Show LUNs**. The switch queries the target to determine which LUN numbers are visible to each configured host.

When you select a specific host, only the LUNs visible to that host are displayed. If you select **All Hosts**, LUNs visible to all configured hosts are displayed. If a LUN is visible to multiple hosts, it is listed once for each host.

7. Select the check box in the **Add** column to add a LUN. You can use the **Select/De-select All** button to add all the LUNs, or to clear all selections.
8. Select the **Current LUN State**, which refers to data already on the LUN.
 - If the LUN is not encrypted, the correct value is **Clear Text**.
 - If the LUN was previously encrypted, select **Encrypted**.
 - If you disable the existing LUN data, the current LUN state setting does not matter.
 - The desired encryption mode.
 - The disposition for Existing Data.

Warning: If the current LUN state is **Clear Text** and the desired state is encrypted, then a first time re-key will occur. If the current LUN state is **Encrypted** and the desired LUN state is **Clear Text**, a re-key will not occur. You may choose **Disable** from the Existing Data list to avoid this, but then all data on the LUN is lost.

When changing an existing LUN to **Clear Text**, the data must be disabled, so it is recommended you back up the LUN's data first using a host-based application.

NOTE

For tape devices, the Existing Data components and the Current LUN State do not display.

9. If you want to enforce a **Re-keying Interval**, enter the number of days that you want to use a key before obtaining a new key. A value of 0 is equivalent to Infinite, which is the default.

The **Re-keying Interval** field is editable only if the LUNs are encrypted. If **Clear Text** is selected as the encryption mode, **Re-Keying Interval** is disabled.

NOTE

For disk LUNs, expiration of the re-keying interval automatically triggers generation of a new key and starts a re-keying operation (reads and re-writes all data on the disk LUN).

10. Click **OK**.
11. Click **Commit** in the **Encryption Targets** dialog box when the LUNs have been added for all hosts that will access them.

NOTE

If there are other hosts that will access the same physical LUNs by way of other target ports (and thus other target containers), add the LUNs for the other hosts before you click **Commit**.

Adding Target Tape LUNs for encryption

You configure a Crypto LUN by adding the LUN to the CryptoTarget container and enabling the encryption property on the Crypto LUN. You must add LUNs manually. After you add the LUNs, you must specify the encryption settings.

When configuring a LUN with multiple paths, the same LUN policies must be configured on all the LUN's paths. If there are multiple paths to the same physical LUNs, then the LUNs are added to multiple target containers (one target per storage device port). See [“Configuring encrypted storage in a multi-path environment”](#) on page 66 for a multi-path configuration scenario.

1. Select **Configure > Encryption** from the menu bar.

The **Encryption Center** dialog box displays.

2. Select the encryption group, switch, or encryption engine containing the storage device to be configured.

3. Click **Encryption Targets**.

The **Encryption Targets** dialog box displays.

4. Select a Target storage device from the list, and click **LUNs**.

The **Encryption Target LUNs** dialog box displays.

5. Click **Add**.

The **Add Encryption Target Tape LUNs** dialog box displays.

This dialog box includes a table of all LUNs in the storage device that are visible to hosts. LUNs are identified by the Host world wide name, LUN number, and Volume Label Prefix number.

6. Select a host from the **Host** list.

Before you encrypt a LUN you must select a host and then either discover LUNs that are visible to the virtual initiator representing the selected host, or enter a range of LUN numbers to be configured for the selected host.

7. Choose a LUN to be added to an encryption target container using one of the two following methods:

- **Discover.** Click to identify the exposed logical unit number for a specified initiator. If you already know the exposed LUNs for the various initiators accessing the LUN, you can enter the range of LUNs using the alternative method.
- **Enter a LUN number range.** Click to add a range of LUNs to be configured for the selected host. The LUN needed for configuring a Crypto LUN is the LUN that is exposed to a particular initiator.

2 Configuring encrypted storage in a multi-path environment

8. Select the desired encryption mode.
 - If you change a LUN policy from **Native Encryption** or **DF-Compatible Encryption** to **Clear Text**, you disable encryption.
 - The LUNs of the target which are not enabled for encryption must still be added to the CryptoTarget container with the **Clear Text** encryption mode option.

NOTE

The Re-keying interval can only be changed for disk LUNs. For tape LUNs, expiration of the re-keying interval simply triggers the generation of a new key, to be used on future tape volumes. Tapes that are already made are not re-keyed. To re-key a tape, you would need to read the tape contents using a host application that decrypts the tape contents using the old key, and then re-write the tape, which re-encrypts the data with the new key.

9. Click **OK**.

The selected tape LUNs are added to the encryption target container.

Configuring encrypted storage in a multi-path environment

This example assumes one host accessing one storage device using two paths:

- The first path is from host port A to target port A, using encryption engine A for encryption.
- The second path is from host port B to target port B, using encryption engine B for encryption.

Encryption engines A and B are in switches that are already part of encryption group X.

The following is the procedure for configuring this scenario using the Management application.

1. Zone host port A and target port A, using the **Configure > Zoning** dialog box.
2. Zone host port B and target port B, using the **Configure > Zoning** dialog box.
3. Open the **Encryption Center** dialog box by selecting **Configure > Encryption** from the Management application's main menu.
4. Click the **View By Encryption Groups** button to display the encryption groups.
5. Select encryption group X, then click the **Encryption Targets** button.
6. Click the **Add** button to start the **Configure Storage Encryption** wizard. Use the **Configure Storage Encryption** wizard to create a target container for encryption engine A with target port A and host port A.
7. Run the **Configure Storage Encryption** wizard again to create a target container for encryption engine B with target port B and host port B.

Up to this point, the Management application has been automatically committing changes as they are made. The targets and hosts are now fully configured; only the LUN configuration remains.

8. In the **Encryption Targets** dialog box, select target port A, click **LUNs**, then click **Add**. Select the LUNs to be encrypted and the encryption policies for the LUNs.

9. Select target port B, click **LUNs**, then click **Add**. Select the LUNs to be encrypted and the encryption policies for the LUNs, making sure that the encryption policies match the policies specified in the other path.
10. Click **Commit** to make the LUN configuration changes effective in both paths simultaneously.

The Management application does not automatically commit LUN configuration changes. This allows matching changes made in a multi-path environment to be committed together, preventing cases where one path may be encrypting and another path is not encrypting, resulting in corrupted data. You must remember to click the **Commit** button after any LUN configuration changes, even in non-multi-path environments. The **Encryption Targets** dialog box displays a reminder if you attempt to close the dialog box without committing LUN configuration changes.

NOTE

There is a limit of 25 uncommitted LUN configuration changes. When adding more than 12 LUNs in a multi-path environment, repeat steps [step 8](#) through [step 10](#) above, adding only 12 LUNs to each target container at a time. Each commit operation, then, will commit 24 LUNs, 12 in each path.

Master keys

When an opaque key vault is used, a master key is used to encrypt the data encryption keys. The master key status indicates whether a master key is used and whether it has been backed up. Encryption is not allowed until the master key has been backed up.

Only the active master key can be backed up, and multiple backups are recommended. You can back up or restore the master key to the key vault, to a file, or to a recovery card set. A recovery card set is set of smart cards. Each recovery card holds a portion of the master key. The cards must be gathered and read together from a card reader attached to a PC running the Brocade SAN Management Application to restore the master key.

NOTE

It is very important to back up the master key because if the master key is lost, none of the data encryption keys can be restored and none of the encrypted data can be decrypted.

Active master key

The active master key is used to encrypt newly-created data encryption keys (DEKs) prior to sending them to a key vault to be stored. You can restore the active master key under the following conditions:

- The active master key has been lost, which happens if all encryption engines in the group have been zeroized or replaced with new hardware at the same time.
- You want multiple encryption groups to share the same active master key. Groups should share the same master key if the groups share the same key vault and tapes (or disks) are going to be regularly exchanged between the groups.

Alternate master key

The alternate master key is used to decrypt data encryption keys that were not encrypted with the active master key. Restore the alternate master key for the following reasons:

- To read an old tape that was created when the group used a different active master key.
- To read a tape (or disk) from a different encryption group that uses a different active master key.

Master key actions

Master key actions are as follows:

- **Backup master key**, which is enabled any time a master key exists.
- **Restore master key**, which is enabled when no master key exists or the previous master key has been backed up.
- **Create new master key**, which is enabled when no master key exists or the previous master key has been backed up.

Reasons master keys can be disabled

Master key actions are disabled if unavailable. There are several ways a master key can be disabled:

- The user does not have Storage Encryption Security permissions. See [“Encryption user privileges”](#) on page 17 for more information.
- The group leader is not discovered or managed by the Management application.

Saving the master key to a file

Use the following procedure to save the master key to a file.

1. Select **Configure > Encryption** from the menu bar.
The **Encryption Center** dialog box displays.
2. Select an encryption group from the tree, and click **Properties**.

NOTE

Master keys belong to the group and are managed from the group properties.

3. Select the **Security** tab.
4. Select **Backup Master Key** as the **Master Key Action**.

The **Master Key Backup** dialog box displays, but only if the master key has already been generated.

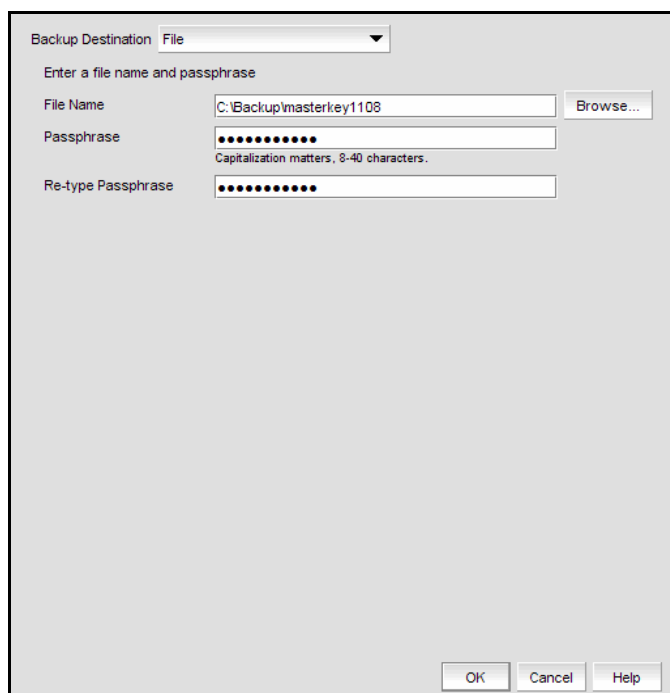


FIGURE 47 Backup Destination (to file) dialog box

5. Select **File** as the **Backup Destination**.
6. Enter a file name, or browse to the desired location.
7. Enter the passphrase, which is required for restoring the master key. The passphrase can be between eight and 40 characters, and any character is allowed.
8. Re-type the passphrase for verification.
9. Click **OK**.

ATTENTION

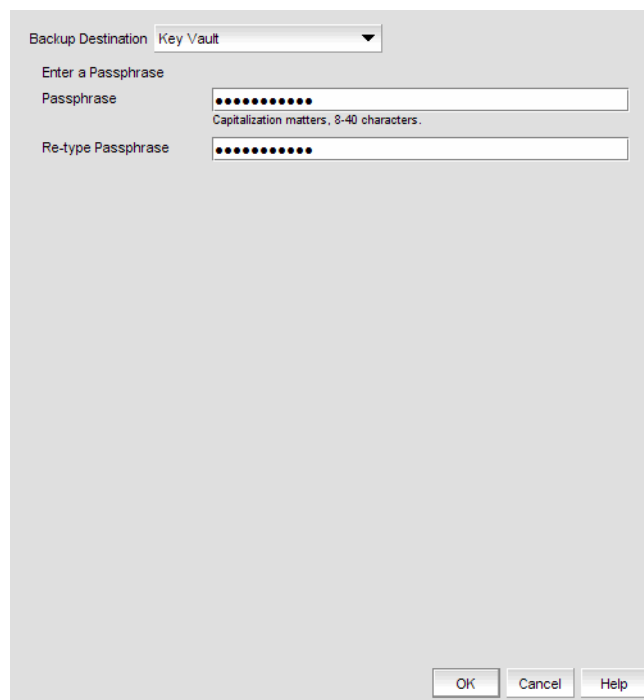
Save the passphrase. This passphrase is required if you ever need to restore the master key from the file.

Saving a master key to a key vault

Use the following procedure to save the master key to a key vault.

1. Select **Configure > Encryption** from the menu bar.
The **Encryption Center** dialog box displays.
2. Select an encryption group from the tree, and click **Properties**.
3. Select the **Security** tab.
4. Select **Backup Master Key** as the **Master Key Action**.

The **Backup Master Key for Encryption Group** dialog box displays.



The dialog box is titled "Backup Master Key for Encryption Group". At the top, there is a dropdown menu labeled "Backup Destination" with "Key Vault" selected. Below this, the text "Enter a Passphrase" is displayed. There are two text input fields. The first field is labeled "Passphrase" and contains a series of dots. Below it, a hint reads "Capitalization matters, 8-40 characters." The second field is labeled "Re-type Passphrase" and also contains a series of dots. At the bottom right of the dialog box, there are three buttons: "OK", "Cancel", and "Help".

FIGURE 48 Backup Destination (to key vault) dialog box

5. Select **Key Vault** as the **Backup Destination**.
6. Enter the passphrase, which is required for restoring the master key. The passphrase can be between eight and 40 characters, and any character is allowed.
7. Re-type the passphrase for verification.
8. Click **OK**.
A dialog box displays that shows the **Key ID**.
9. Store both the **Key ID** and the passphrase in a secure place. Both will be required to restore the master key in the future. (The **Key ID** identifies the storage location in the key vault.)
10. Click **OK** after you have copied the key ID.

Saving a master key to a smart card set

A card reader must be attached to the SAN Management application PC to complete this procedure. Recovery cards can only be written once to back up a single master key. Each master key backup operation requires a new set of previously unused smart cards.

NOTE

Windows operating systems do not require smart card drivers to be installed separately; the driver is bundled with the operating system. You must install a smart card driver for Linux and Solaris operating systems, however. For instructions, see the *Data Center Fabric Manager Administrator's Guide*.

The key is divided between the cards in the card set. When the master key is backed up to a set of three cards, a minimum of two cards can be used together to restore the master key. When the master key is backed up to a set of five cards, a minimum of three cards can be used together to restore the master key. Backing up the master key to multiple recovery cards is the recommended and most secure option.

NOTE

When you write the key to the card set, be sure you write the full set without canceling. If you cancel, all the previously written cards become unusable, and you will need to discard them and create a new set.

-
1. Select **Configure > Encryption** from the menu bar.
The **Encryption Center** dialog box displays.
 2. Select an encryption group from the tree, and click **Properties**.
 3. Select the **Security** tab.
 4. Select **Backup Master Key** as the **Master Key Action**.
The **Backup Master Key for Encryption Group** dialog box displays.

FIGURE 49 Backup Destination (to smart cards) dialog box

5. Select **A Recovery Set of Smart Cards** as the **Backup Destination**.
6. Enter the recovery card set size.
7. Insert the first blank card and wait for the card serial number to appear.
8. Run the additional cards needed for the set through the reader. As you read each card, the card ID displays in the **Card Serial#** field. Be sure to wait for the ID to appear.
9. Enter the mandatory last name and first name of the person to whom the card is assigned.
10. Type a Card **Password**.
11. Re-type the password for verification.
12. Record and store the password in a secure location.
13. Click **Write Card**.
The dialog box prompts you to insert the next card, up to the number of cards specified in [step 6](#).
14. Repeat [step 7](#) through [step 13](#) for each card.
15. Continue until you have written to all the cards in the set.
16. After the last card is written, click **OK** in the **Master Key Backup** dialog box to finish the operation.

Restoring a master key from a file

Use the following procedure to restore the master key from a file.

1. Select **Configure > Encryption** from the menu bar.
The **Encryption Center** dialog box displays.
2. Select an encryption group from the tree, and click **Properties**.
3. Select the **Security** tab.
4. Select **Restore Master Key** as the **Master Key Action**.

The **Restore Master Key for Encryption Group** dialog box displays.

Select a Master Key to Restore

Active Master Key - The resulting key will be used for all new data encryption.

Alternate Master Key - The resulting key can be used for reading old tapes.

Restore From: File

Enter a file name and passphrase

File name: Browse...

Passphrase:
Capitalization matters, 8-40 characters.

OK Cancel Help

FIGURE 50 Select a Master Key to Restore (from file) dialog box

5. Choose the active or alternate master key for restoration, as appropriate. Refer to [“Active master key”](#) on page 67 and [“Alternate master key”](#) on page 68 if you need more information on active and alternate master keys.
6. Select **File** as the **Restore From** location.
7. Enter a file name, or browse to the desired location.
8. Enter the passphrase. The passphrase that was used to back up the master key must be used to restore the master key.
9. Click **OK**.

Restoring a master key from a key vault

Use the following procedure to restore the master key from a key vault.

1. Select **Configure > Encryption** from the menu bar.
The **Encryption Center** dialog box displays.
2. Select an encryption group from the tree, and click **Properties**.
3. Select the **Security** tab.
4. Select **Restore Master Key** as the **Master Key Action**.

The **Restore Master Key for Encryption Group** dialog box displays.

Select a Master Key to Restore

Active Master Key - The resulting key will be used for all new data encryption.

Alternate Master Key - The resulting key can be used for reading old tapes.

Restore From: Key Vault

Key ID:

Enter a passphrase to decrypt the master key

Passphrase:

Capitalization matters, 8-40 characters.

OK Cancel Help

FIGURE 51 Select a Master Key to Restore (from key vault) dialog box

5. Choose the active or alternate master key for restoration, as appropriate. Refer to [“Active master key”](#) on page 67 and [“Alternate master key”](#) on page 68 if you need more information on active and alternate master keys.
6. Select **Key Vault** as the **Restore From** location.
7. Enter the key ID of the master key that was backed up to the key vault.
8. Enter the passphrase. The passphrase that was used to back up the master key must be used to restore the master key.
9. Click **OK**.

Restoring a master key from a smart card set

A card reader must be attached to the SAN Management application PC to complete this procedure.

Use the following procedure to restore the master key from a set of smart cards.

1. Select **Configure > Encryption** from the menu bar.
The **Encryption Center** dialog box displays.
2. Select an encryption group from the tree, and click **Properties**.
3. Select the **Security** tab.
4. Select **Restore Master Key** as the **Master Key Action**.

The **Restore Master Key for Encryption Group** dialog box displays.

FIGURE 52 Select a Master Key to Restore (from a recovery set of smart cards) dialog box

5. Choose the active or alternate master key for restoration, as appropriate. Refer to [“Active master key”](#) on page 67 and [“Alternate master key”](#) on page 68 if you need more information on active and alternate master keys.
6. Select **A Recovery Set of Smart Cards** as the **Restore From** location.
7. Insert the recovery card containing a share of the master key that was backed up earlier, and wait for the card serial number to appear.
8. Enter the password that was used to create the card. After five unsuccessful attempts to enter the correct password, the card becomes locked and unusable.
9. Click **Restore**.
The dialog box prompts you to insert the next card, if needed.
10. Insert the next card, and repeat [step 8](#) and [step 9](#).

11. Continue until all the cards in the set have been read.
12. Click **OK**.

Creating a new master key

Though it is generally not necessary to create a new master key, you may be required to create one due to circumstances such as the following:

- The previous master key has been compromised.
- Corporate policy might require a new master key every year for security purposes.

When you create a new master key, the former active master key automatically becomes the alternate master key.

The new master key cannot be used (no new data encryption keys can be created, so no new encrypted LUNs can be configured), until you back up the new master key. After you have backed up the new master key, it is strongly recommended that all encrypted disk LUNs be re-keyed. Re-keying causes a new data encryption key to be created and encrypted using the new active master key, thereby removing any dependency on the old master key.

1. Select **Configure > Encryption**.
2. Select an encryption group from the tree and click **Properties**.
3. Select the **Security** tab.
4. Select **Create a New Master Key** from the list.

The **Confirm Master Key Creation** dialog box displays.

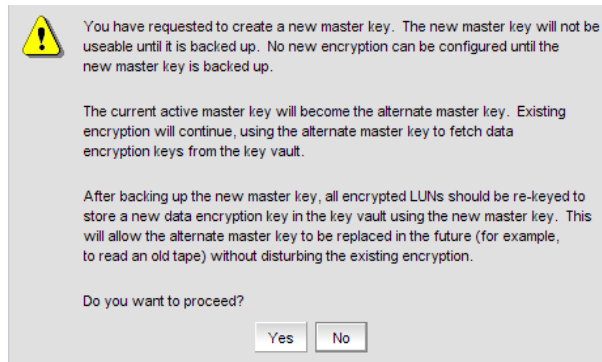


FIGURE 53 Confirm master key creation dialog box

5. Read the information, and click **Yes** to proceed.

Zeroizing an encryption engine

Zeroizing is the process of erasing all data encryption keys and other sensitive encryption information in an encryption engine. You can zeroize an encryption engine manually to protect encryption keys. No data is lost because the data encryption keys for the encryption targets are stored in the key vault.

Zeroizing has the following effects:

- All copies of data encryption keys kept in the encryption switch or encryption blade are erased.
- Internal public and private key pairs that identify the encryption engine are erased and the encryption switch or the encryption blade is in the FAULTY state.
- All encryption operations on this engine are stopped and all virtual initiators (VI) and virtual targets (VT) are removed from the fabric's name service.
- The key vault link key (for NetApp LKM key vaults) or the master key (for other key vaults) is erased from the encryption engine.

Once enabled, the encryption engine is able to restore the necessary data encryption keys from the key vault when the link key (for the NetApp Lifetime Key Management application) or the master key (for other key vaults) are restored.

- If the encryption engine was part of an HA cluster, targets fail over to the peer which assumes the encryption of all storage targets. Data flow will continue to be encrypted.
- If there is no HA backup, host traffic to the target will fail as if the target has gone offline. The host will not have unencrypted access to the target. There will be no data flow at all because the encryption virtual targets will be offline.

NOTE

Zeroizing an engine affects the I/Os but all target and LUN configuration is intact. Encryption target configuration data is not deleted.

You can zeroize an encryption engine only if it is enabled (running) or disabled, but ready to be enabled. If the encryption engine is not in one of these states, an error message displays.

When using a NetApp LKM key vault, if all the encryption engines in a switch are zeroized, the switch loses the link key required to communicate with the LKM vault. After the encryption engines are rebooted and re-enabled, you must use the CLI to create new link keys for the switch.

When using an opaque key vault, if all the encryption engines in an encryption group are zeroized, the encryption group loses the master key required to read data encryption keys from the key vault. After the encryption engines are rebooted and re-enabled, you must restore the master key from a backup copy, or alternatively you can also generate a new master key and back it up. Restoring the master key from a backup copy or generating a new master key and backing it up indicates that all previously generated DEKs will not be decryptable, unless the original master key used to encrypt them is restored.

Use the **Restore Master key** wizard from the **Encryption Group Properties** dialog box to restore the master key from a backup copy.

1. Select **Configure > Encryption** from the menu bar.

The **Encryption Center** dialog box displays.

2. Select the encryption engine, and then click **Zeroize**.

A confirmation dialog box describing consequences and actions required to recover launches.

2 Zeroizing an encryption engine

3. Initialize the encryption engine.

An automatic power cycle and reboot occurs on the encryption blade and encryption switch.

4. Enable the encryption engine using the **Switch Encryption Properties** dialog box:

- a. Select the encryption engine from the **Encryption Center** dialog box.
- b. Click the **Properties** button.

The **Switch Encryption Properties** dialog box displays.

Switch Properties		Property Value
Name		MMA-7-57-SW
Node WWN		10:00:00:05:1E:53:FB:E3
Switch Status	⚠ Marginal	
Switch Membership Status		Group Leader
Encryption Group		routingTestGrp
Encryption Group Status		OK - Converged
Fabric		10:00:00:05:1E:53:FB:E3
Domain ID		1
Firmware Version		v6.2.0v6.2.0_pit_a_081104_2000
Key Vault Type		RSA Key Manager (RKM)
Primary Key Vault Link Key Status		Not Used
Primary Key Vault Connection Status		Failed authentication
Backup Key Vault Link Key Status		Not Used
Backup Key Vault Connection Status		Key Vault Not Configured

Public Key Certificate

Version: V3
 Subject: OU=Technical Support, O=BRCD, L=San Jose, ST=CA, C=US, CN=krac.000000051e53fbc3
 Signature Algorithm: SHA1 withRSA, OID = 1.2.840.113549.1.1.5
 Key: Sun RSA public key, 4096 bits
 modulus:
 793025492310457750649467392752126268798013306411578746455995553106479629614175625685807770584
 40442668701000435344220240004370320004466846442545400437000004420434450034003750004455700

Encryption Engine Properties		Engine
Current Status	⚠ Zeroized	
Set State To	Enabled (New State)	
Encrypted Targets	0	
HA Cluster Peer	No Peer	
HA Cluster Name	No Cluster	

OK Cancel Help

FIGURE 54 Switch Encryption Properties dialog box

- c. Select **Enabled (New State)** from the **Set State To** list for each encryption engine.
- d. Click **OK**.

Tracking Smart Cards

Smart Cards, which are credit card-sized cards that contain a CPU and persistent memory, are a secure way to back up and restore a master key. Using Smart Cards is optional. Master keys can also be backed up to a file or key vaults and are only used for encryption groups using RKM or HP SKM key vaults.

Even if an encryption group is deleted, the smart cards are still displayed. You must manually delete them.

Use the **Smart Card Asset Tracking** dialog box to track Smart Card details.

1. Select **Configure > Encryption** from the menu bar.

The **Encryption Center** dialog box displays.

2. Click **Smart Card Tracking**.

The **Smart Card asset tracking** dialog box displays.

Known smart cards are listed in the first table. Select a card to display its details in the lower table.

Card ID	Card Type	Usage	First Name	Last Name	Notes
---------	-----------	-------	------------	-----------	-------

Remove Save As...

Card Details

OK Cancel Help

FIGURE 55 Smart Card asset tracking dialog box

Clicking the **Remove** button removes a selected smart card from the Management application database. You can remove smart cards to keep the **Smart Cards** table at a manageable size, but removing the card from the table does not invalidate it. The Smart Card can still be used.

Clicking the **Save As** button saves the entire list of smart cards to a file. The available formats are comma-separated values (.csv) and HTML files (.html).

Encryption-related acronyms in log messages

Fabric OS log messages related to encryption components and features may have acronyms embedded that require interpretation. [Table 5](#) lists some of those acronyms.

TABLE 5 Encryption Acronyms

Acronym	Name
EE	Encryption Engine
EG	Encryption Group
HAC	High Availability Cluster

Encryption configuration using the CLI

In this chapter

• Overview	81
• Command validation checks	82
• Command RBAC permissions and AD types	83
• Cryptocfg Help command output	86
• Setting default zoning to no access	87
• Management port configuration	87
• I/O sync link configuration	88
• Encryption switch initialization	90
• Basic encryption group configuration	95
• Key vault configuration	99
• High Availability (HA) cluster configuration	100
• CryptoTarget container configuration	102
• Crypto LUN configuration	109
• Configuring a multi-path Crypto LUN	117
• Tape pool configuration	120
• Data re-keying	125
• First time encryption	129

Overview

This chapter explains how to use the command line interface (CLI) to configure a Brocade Encryption Switch, or an FS8-18 Encryption blade in a DCX or DCX-4S to perform data encryption.

This chapter assumes that the basic setup and configuration of the Brocade Encryption Switch, DCX, or DCX-4S has been done as part of the initial hardware installation, including setting the management port IP address.

For command syntax and description of parameters, refer to the *Fabric OS Command Reference Manual, v6.3.0*.

NOTE

The configuration tasks described in this chapter build and depend on each other and should be performed in sequence to avoid unnecessary errors.

Command validation checks

Before a command is executed, it is validated against the following checks.

1. Active or Standby availability: on enterprise-class platforms, checks that the command is available on the Control Processor (CP).
2. Role Based Access Control (RBAC) availability: checks that the invoking user's role is permitted to invoke the command. If the command modifies system state, the user's role must have *modify* permission for the command. If the command only displays system state, the user's role must have *observe* permission for the command. Some commands both observe and modify system state and thus require *observe-modify* permission. The following RBAC permissions are supported:
 - O = observe
 - OM = observe-modify,
 - N = none/not available
3. Admin Domain availability: checks that the command is allowed in the currently selected Admin Domain. For information on Admin Domain concepts and restrictions, refer to the *Fabric OS Administrator's Guide*.

Admin Domain Types are one or more of the following. If more than one AD type is listed for a command, the AD type is option-specific. Display options may be allowed, but set options may be subject to Admin Domain restrictions.

- SwitchMember = Allowed to execute only if the local switch is part of the current AD.
 - Allowed = Allowed to execute in all ADs.
 - PhysFabricOnly = Allowed to execute only in AD255 context (and the user should own access to AD0-AD255 and have admin RBAC privilege).
 - Disallowed = Only allowed to execute in AD0 or AD255 context, not allowed in AD1-AD254 context.
 - PortMember = All control operations allowed only if the port or the local switch is part of the current AD. View access allowed if the device attached to the port is part of the current AD.
 - AD0Disallowed = Allowed to execute only in AD255 and AD0 (if no ADs are configured).
 - AD0Only = Allowed to execute only in AD0 when ADs are not configured.
4. Command-specific: checks whether the command is supported on the platform for which it is targeted.

Command RBAC permissions and AD types

There are two RBAC roles that are permitted to perform Encryption operations.

1. Admin and SecurityAdmin

Users authenticated with the Admin and SecurityAdmin RBAC roles may perform cryptographic functions assigned to the FIPS Crypto Officer including the following:

- Perform encryption node initialization.
- Enable cryptographic operations.
- Manage input/output functions of critical security parameters (CSPs).
- Zeroize encryption CSPs.
- Register and configure a key vault.
- Configure a recovery share policy.
- Create and register recovery share.
- Perform encryption group- and clustering-related operations.
- Manage keys, including creation, recovery, and archiving functions.

2. Admin and FabricAdmin

Users authenticated with the Admin and FabricAdmin RBAC roles may perform routine Encryption Switch management functions including the following:

- Configure virtual devices and crypto LUNs.
- Configure LUN and tape associations.
- Perform re-keying operations.
- Perform firmware download.
- Perform regular Fabric OS management functions.

Refer to [Table 6](#) for the RBAC permissions of the encryption configuration commands.

TABLE 6 Encryption command RBAC availability and admin domain type¹

Command name	User	Admin	Operator	Switch Admin	Zone Admin	Fabric Admin	Basic Switch Admin	Security Admin	Admin Domain
addgroupmember	N	OM	N	N	N	O	N	OM	Disallowed
addmembemode	N	OM	N	N	N	O	N	OM	Disallowed
addhaclustermember	N	OM	N	N	N	OM	N	O	Disallowed
addinitiator	N	OM	N	N	N	OM	N	O	Disallowed
addLUN	N	OM	N	N	N	OM	N	O	Disallowed
commit	N	OM	N	N	N	OM	N	O	Disallowed
createcontainer	N	OM	N	N	N	OM	N	O	Disallowed
createencgroup	N	OM	N	N	N	O	N	OM	Disallowed
createhacluster	N	OM	N	N	N	OM	N	O	Disallowed

3 Command RBAC permissions and AD types

TABLE 6 Encryption command RBAC availability and admin domain type¹ (Continued)

Command name	User	Admin	Operator	Switch Admin	Zone Admin	Fabric Admin	Basic Switch Admin	Security Admin	Admin Domain
createtapepool	N	OM	N	N	N	OM	N	O	Disallowed
deletecontainer	N	OM	N	N	N	OM	N	O	Disallowed
deleteencgroup	N	OM	N	N	N	O	N	OM	Disallowed
deletefile	N	OM	N	N	N	O	N	OM	Disallowed
deletehacluster	N	OM	N	N	N	OM	N	O	Disallowed
deletetapepool	N	OM	N	N	N	OM	N	O	Disallowed
dereggroupleader	N	OM	N	N	N	O	N	OM	Disallowed
deregkeyvault	N	OM	N	N	N	O	N	OM	Disallowed
deregmembermode	N	OM	N	N	N	O	N	OM	Disallowed
dhchallenge	N	OM	N	N	N	O	N	OM	Disallowed
dhresponse	N	OM	N	N	N	O	N	OM	Disallowed
disableEE	N	OM	N	N	N	O	N	OM	Disallowed
discoverLUN	N	OM	N	N	N	OM	N	O	Disallowed
eject	N	OM	N	N	N	O	N	OM	Disallowed
enable	N	OM	N	N	N	O	N	OM	Disallowed
enableEE	N	OM	N	N	N	O	N	OM	Disallowed
export	N	OM	N	N	N	O	N	OM	Disallowed
exportmasterkey	N	OM	N	N	N	O	N	OM	Disallowed
fallback	N	OM	N	N	N	OM	N	O	Disallowed
genmasterkey	N	OM	N	N	N	O	N	OM	Disallowed
help	N	OM	N	N	N	O	N	OM	Disallowed
import	N	OM	N	N	N	O	N	OM	Disallowed
initEE	N	OM	N	N	N	O	N	OM	Disallowed
initnode	N	OM	N	N	N	O	N	OM	Disallowed
leave_encryption_group	N	OM	N	N	N	O	N	OM	Disallowed
manual_rekey	N	OM	N	N	N	OM	N	O	Disallowed
modify	N	OM	N	N	N	OM	N	O	Disallowed
move	N	OM	N	N	N	OM	N	O	Disallowed
recovermasterkey	N	OM	N	N	N	O	N	OM	Disallowed
regEE	N	OM	N	N	N	O	N	OM	Disallowed

TABLE 6 Encryption command RBAC availability and admin domain type¹ (Continued)

Command name	User	Admin	Operator	Switch Admin	Zone Admin	Fabric Admin	Basic Switch Admin	Security Admin	Admin Domain
reggroupleader	N	OM	N	N	N	O	N	OM	Disallowed
regkeyvault	N	OM	N	N	N	O	N	OM	
regmembermode	N	OM	N	N	N	O	N	OM	
removehaclustermember	N	OM	N	N	N	OM	N	O	
removeinitiator	N	OM	N	N	N	OM	N	O	
removeLUN	N	OM	N	N	N	OM	N	O	
replace	N	OM	N	N	N	OM	N	O	
set	N	OM	N	N	N	O	N	OM	
setEE	N	OM	N	N	N	O	N	OM	
show	N	OM	N	N	N	O	N	OM	
transabort	N	OM	N	N	N	OM	N	O	
transshow	N	OM	N	N	N	OM	N	O	
zeroizeEE	N	OM	N	N	N	O	N	OM	

1. Legend: O = observe, OM = observe-modify, N = none/not available

Cryptocfg Help command output

All encryption operations are done using the **cryptocfg** command. The **cryptocfg** command has an help output that lists all options.

```
switch:admin> cryptocfg --help
Usage: cryptocfg
--help -nodecfg:
    Display the synopsis of node parameter configuration.
--help -groupcfg:
    Display the synopsis of group parameter configuration.
--help -hacluster:
    Display the synopsis of hacluster parameter configuration.
--help -devicecfg:
    Display the synopsis of device container parameter configuration.
--help -transcfg:
    Display the synopsis of transaction management.

switch:admin> cryptocfg --help -nodecfg
Usage: cryptocfg
--help -nodecfg:
    Display the synopsis of node parameter configuration.
--initnode:
    Initialize the node for configuration of encryption options.
--initEE [<slotnumber>]:
    Initialize the specified encryption engine.
--regEE [<slotnumber>]:
    Register a previously initialized encryption blade.
--reg -membertnode <member node WWN> <member node certfile> <IP addr>:
    Register a member node with the system.
--reg -groupleader <group leader WWN> <group leader certfile> <IP addr>:
    Register a group leader node with the system.
```


Setting default zoning to no access

Initially, default zoning for all Brocade switches is set to All Access. This is generally the default zoning setting within a fabric. The All Access setting allows the Brocade Encryption Switch, DCX, or DCX-4S to join the fabric (If there is a difference in this setting within the fabric, the fabric will segment).

Before committing an encryption configuration in a fabric, default zoning must be set to No Access within the fabric. When encryption is implemented, frames sent between a host and a target LUN are redirected to a virtual target within an encryption switch or blade. Redirection zones are created to route these frames. When redirection zones are in effect, direct access from host to target should not be allowed to prevent data corruption. The No Access setting ensures that no two devices on the fabric can communicate with one another without going through a regular zone or a redirection zone.

1. Check the default zoning setting. Commonly, it will be set to All Access.

```
switch:admin> defzone --show
Default Zone Access Mode
    committed - All Access
    transaction - No Transaction
```

2. From any configured primary FCS switch, change the default zoning setting to No Access.

```
switch:admin> defzone --noaccess
switch:admin> cfgfsave
```

The change will be applied within the entire fabric.

Management port configuration

Each encryption switch has one GbE management port. In the case of a DCX or DCX-4S with FS8-18 blades installed, management ports are located on the CP blades. The management port connects to the key management system and optionally to DCFM. All switches you plan to include in an encryption group must be connected to the same dedicated LAN management network. All nodes within an encryption group, the key management system, and DCFM must have like IP settings (all IPv4 or all IPv6) on their management interfaces. To eliminate DNS traffic and potential security risks related to DHCP, DHCP should not be used. A static IP address should be assigned.

I/O sync link configuration

Each encryption switch or FS8-18 blade has two GbE ports labeled Ge0 and Ge1. The Ge0 and Ge1 ports connect encryption switches and FS8-18 blades to other encryption switches and FS8-18 blades. These two ports provide link layer redundancy rather than being used for the IP network redundancy. The the ports are bonded together as a single virtual network interface, and are collectively referred to as the I/O sync link. Only one IP address is used. All encryption switches or blades must be interconnected by their I/O sync links through a dedicated LAN. Both ports of each encryption switch or blade must be connected to the same IP network, and the same subnet. Avoid VLANs, if possible. To eliminate DNS traffic and potential security risks related to DHCP, DHCP should not be used. Static IP addresses should be assigned.

The IP address of the I/O sync link must be configured before enabling the encryption engine for encryption. If the IP address is configured after the encryption engine is enabled for encryption, the encryption switch needs to be rebooted, and the encryption blade needs to be powered off and powered on (slotpoweroff/slotpoweron) for the IP address configuration to take effect. The configured GE Ports must be connected to the network when deploying an encryption switch or blade in an encryption group before performing any Re-Key operations. Failure to do so will result in Re-Key operation not starting in the encryption group or high availability (HA) cluster.

If the IP address of the I/O sync link ports is modified after encryption engine is enabled for encryption, the encryption switch needs to be rebooted, and the encryption blade needs to be powered off and powered on (slotpoweroff/slotpoweron) for the modified IP address to take effect. Failure to do so will result in Re-Key operations not starting in the encryption group or high availability (HA) cluster.

Assigning static IP addresses to Ge0 and Ge1

The Ge0 and Ge1 ports are bonded together as a single virtual network interface that provides link layer redundancy. Only Ge0 needs to be configured. Always use **ipaddrset -eth0** to configure the address. If an address is assigned to ge1 (-eth1), it is accepted and stored, but it is ignored. The Ge0 and Ge1 addresses must be configured before initializing the encryption switch or blade.

1. Log into the switch as Admin or FabricAdmin.
2. Configure the IP address using the **ipaddrset** command. Only IPv4 addresses are supported.
Only -eth0 needs to be configured. Always use -eth0. The following example configures a static IP address and gateway address for the bonded interface.

```
switch:admin> ipaddrset -eth0 --add 10.32.33.34/23
switch:admin> ipaddrset -gate --add 10.32.1.1
```

Special consideration for blades

For FS8-18 blades, the slot number must also be included in the **ipaddrset** command, for example:

```
switch:admin> ipaddrset -slot 7 -eth0 --add 10.32.33.34/23
switch:admin> ipaddrset -slot 7 -gate --add 10.32.1.1
```

There are additional considerations if blades are removed and replaced, or moved to a different slot. On chassis-based systems, IP addresses are assigned to the slot rather than the blade, and are saved in non-volatile storage on the control processor blades. IP addresses may be assigned even if no blade is present. If an FS8-18 blade is installed in a slot that was previously configured for a different type of blade with two IP ports (an FC4-16E blade, for example), the FS8-18 blade is assigned the address specified for -eth0 in that slot.

To be sure the correct IP addresses are assigned, use the **ipaddrshow** command to display the IP address assignments as shown in the following example.

```
switch:admin> ipaddrshow -slot 7

SWITCH
Ethernet IP Address: 10.33.54.207
Ethernet Subnetmask: 255.255.240.0
Fibre Channel IP Address: none
Fibre Channel Subnetmask: none
Gateway IP Address: 10.33.48.1
DHCP: Off
eth0: 10.33.54.208/20
eth1: none/none
Gateway: 10.33.48.1
```

NOTE

If you modify the IP address of the GbE ports (I/O sync links) after encryption is enabled on the switch, you must reboot the encryption switch or issue a **slotpoweroff/slotpoweron** for the IP address change to take effect. Failure to do so will prevent re-key operations from functioning in the HA cluster or encryption group.

IP Address change of a node within an encryption group

Modifying the IP Address of a node that is part of an encryption group is disruptive in terms of cluster operation. The change causes the encryption group to split and if the node was part of an HA cluster, failover/failback capability is lost. Note that the **ipaddrset** command issues no warning or prevents you from changing a node IP address that is part of a configured encryption group or HA cluster. Follow the steps below to recover from the situation. Note that this recovery does not affect existing host encryption I/O.

- If the node is the group leader, delete and recreate the encryption group with the node that now has a new IP address. Refer to the sections [“Deleting an encryption group”](#) on page 165 and [“Creating an encryption group”](#) on page 95 for instructions.
- If the node is a member node, perform the following steps:
 1. Log into the Group Leader as Admin or SecurityAdmin.
 2. Eject and then de-register the node from the encryption group. Refer to the section [“Removing a node from an encryption group”](#) on page 163 for instructions.
 3. Reregister the node with the group leader and add the member node with new IP address to the encryption group. Refer to the section [“Basic encryption group configuration”](#) on page 95 for instructions.

Encryption switch initialization

When setting up a Brocade Encryption Switch or FS8-18 blade for the first time during deployment for encryption services, and before encryption can be enabled on the switch or blade, you must perform a series of initialization steps. These steps are performed only once and must be executed in the order indicated below. Initialization must be performed on every node that is expected to perform encryption within the fabric.

A node is a Brocade Encryption Switch or a DCX or DCX-4S chassis containing one or more FS8-18 encryption blades. A node is identified by the switch IP address and by the switch WWN, which is subsequently referred to as the node WWN.

NOTE

The initialization process overwrites any authentication data and certificates that reside on the node and the security processor. This means that existing key encryption keys (KEKs) such as link keys or master keys are erased. If this is not a first-time initialization, make sure to export the master key by running `cryptocfg --exportmasterkey` and `cryptocfg --export -scp --currentMK` before running `--initEE`. If this encryption engine was configured with an LKM key vault, you will need to reconfigure the key vault to regenerate the Trusted Link after running `cryptocfg --initEE`. Refer to the section “[Key vault configuration](#)” on page 99 for instructions.

Initializing an encryption switch

Take the following steps to initialize an encryption switch or blade.

1. Log into the switch as Admin or SecurityAdmin.
2. Zeroize all critical security parameters (CSPs) on the switch by entering the `cryptocfg --zeroizeEE` command. Provide a slot number if the encryption engine is a blade.

```
SecurityAdmin:switch>cryptocfg --zeroizeEE
This will zeroize all critical security parameters
ARE YOU SURE (yes, y, no, n): [no]y
Operation succeeded.
```

3. Zeroization leaves the switch or blade faulted. Perform the appropriate action depending on whether the encryption engine is a switch or a blade.
 - When the encryption engine is a Brocade Encryption Switch, reboot the switch.
 - When the encryption engine is an FS8-18 blade, issue the `slotpoweroff slot number` command followed by the `slotpoweron slot number` command.
4. As needed, adjust the date and time of the node to be in sync with the encryption group that it's being added to prior to initializing the node.

NOTE

Changing the date or time to a date or time earlier than was in effect at node initialization will invalidate all certificates and will cause key vault operations to fail for that member node.

5. Initialize the node by entering the `cryptocfg --initnode` command. This step is *not* necessary when adding a new blade to a DCX or DCX-4S consisting of previously configured encryption engines. Successful execution generates the following security parameters and certificates:
 - Node CP certificate
 - Key Archive Client (KAC) certificate

NOTE

Node initialization overwrites any existing authentication data on the node.

```
SecurityAdmin:switch>cryptocfg --initnode
This will overwrite all identification and authentication data
ARE YOU SURE (yes, y, no, n): [no] y
```

```
Notify SPM of Node Cfg
Operation succeeded.
```

6. Initialize the encryption engine by entering the **cryptocfg --initEE** command. Provide a slot number if the encryption engine is a blade. This step generates critical security parameters (CSPs) and certificates in the CryptoModule's security processor (SP). The CP and the SP perform a certificate exchange to register respective authorization data.

```
SecurityAdmin:switch>cryptocfg --initEE
This will overwrite previously generated identification
and authentication data
ARE YOU SURE (yes, y, no, n): y
Operation succeeded.
```

7. Register the encryption engine by entering the **cryptocfg --regEE** command. Provide a slot number if the encryption engine is a blade. This step registers the encryption engine with the CP or chassis. Successful execution results in a certificate exchange between the encryption engine and the CP through the FIPS boundary.

```
SecurityAdmin:switch>cryptocfg --regEE
Operation succeeded.
```

NOTE

You should complete the encryption group configuration and key vault registration before you enable the Brocade Encryption Switch or the FS8-18 blade for encryption.

8. Enable the encryption engine by entering the **cryptocfg --enableEE** command. Provide a slot number if the encryption engine is a blade.

NOTE

Every time a Brocade Encryption Switch or DCX or DCX-4S chassis containing one or more FS8-18 blade goes through power cycle event, or after issuing **slotpoweroff <slot number>** followed by **slotpoweron <slot number>** for an FS8-18 blade in DCX or DCX-4S Chassis, the encryption engine must be enabled manually by the Security Administrator. Hosts cannot access the storage LUNs through the storage paths exposed on this Brocade Encryption Switch or FS8-18 blade until the encryption engine is enabled. The encryption engine state can be viewed using the **cryptocfg --show -localEE** command, or by displaying switch or blade properties from DFCM. An encryption engine that is not enabled indicates **Waiting for Enable EE**.

```
SecurityAdmin:switch>cryptocfg --enableEE
Operation succeeded.
```

Checking encryption engine status

You can verify the encryption engine status at any point in the setup process and get information about the next required configuration steps or to troubleshoot an encryption engine that behaves in unexpected ways. Use the `cryptocfg --show -localEE` command to check the encryption engine status.

```
SecurityAdmin:switch>cryptocfg --show -localEE

    EE Slot:                0
      SP state:              Waiting for initEE
      EE key status not available: SP TLS connection is not up.
      No HA cluster membership
EE Slot:                    1
      SP state:              Online
      Current Master KeyID:
a3:d7:57:c7:54:66:65:05:61:7a:35:2c:59:af:a5:dc
      Alternate Master KeyID:
e9:e4:3a:f8:bc:4e:75:44:81:35:b8:90:d0:1f:6f:4d
      HA Cluster Membership: hacDcx2
      EE Attributes:
          EE Route Mode    :    SHARED
          Media Type       :    DISK
EE Slot:                    3
      SP state:              Online
      Current Master KeyID:
a3:d7:57:c7:54:66:65:05:61:7a:35:2c:59:af:a5:dc
      Alternate Master KeyID:
e9:e4:3a:f8:bc:4e:75:44:81:35:b8:90:d0:1f:6f:4d
      No HA cluster membership
      EE Attributes:
          EE Route Mode    :    SHARED
          Media Type       :    DISK
EE Slot:                    10
      SP state:              Online
      Current Master KeyID:
a3:d7:57:c7:54:66:65:05:61:7a:35:2c:59:af:a5:dc
      Alternate Master KeyID:
e9:e4:3a:f8:bc:4e:75:44:81:35:b8:90:d0:1f:6f:4d
      No HA cluster membership
      EE Attributes:
          EE Route Mode    :    SHARED
          Media Type       :    DISK
EE Slot:                    12
      SP state:              Online
      Current Master KeyID:
a3:d7:57:c7:54:66:65:05:61:7a:35:2c:59:af:a5:dc
      Alternate Master KeyID:
e9:e4:3a:f8:bc:4e:75:44:81:35:b8:90:d0:1f:6f:4d
      HA Cluster Membership: hacDcx3
      EE Attributes:
          EE Route Mode    :    SHARED
          Media Type       :    DISK
```

This command provides useful information in the following cases:

- After issuing `initnode`
- After issuing `initEE`.

- After issuing **regEE**.
- After issuing **enableEE**.
- After power cycling an FS8-18 blade.
- After power cycling a DCX or DCX-4S with one or more FS8-18 blades
- To diagnose a “split group” condition where the encryption group status shows DEGRADED but the encryption engine shows online status. Refer to the section “[Encryption group merge and split use cases](#)” on page 171 for more information.

Refer to Appendix A, [Table 17](#) on page 189 for an explanation of encryption engine states (SP states). Refer to Appendix, A, [Table 18](#) on page 190 for an explanation of key encryption (KEK) states.

Certificate Exchange

During the initialization phase a set of RSA key pairs and certificates are generated on every node. These certificates are used for mutual identification and authentication with other group members or with external devices such as key vaults. Every device must have a certificate in order to participate in a deployment of encryption services. Some devices must have each other’s certificates in order to communicate.

Certificates must be exchanged between the key management system you are using and the encryption switch to enable mutual authentication. You must obtain a certificate from the key manager, and import it into the encryption group leader. The encryption group leader exports the certificate to other encryption group members.

A certificate signing request (CSR) must be exported from each switch or blade to an external server or to an attached USB device for signing. The signed certificate must be imported into the switch or blade that generated the CSR, and also must be made available to the key manager. Refer to [Appendix D, “Supported Key Management Systems”](#) for specific procedures.

Exporting a certificate

1. Log into the switch on which the certificate was generated as Admin or SecurityAdmin.
2. Export the certificate from the local switch to an SCP-capable external host or to a mounted USB device. The target server must be SCP-enabled. Enter the **cryptocfg --export** command with the appropriate parameters.

The following example exports a CP certificate from an encryption group member to an external SCP-capable host.

```
SecurityAdmin:switch>cryptocfg --export -scp CPcert \  
192.168.38.245 mylogin /tmp/certs/enc_switch1_cp_cert.pem  
Password:  
Operation succeeded.
```

The following example exports a KAC certificate from the local node to USB storage.

```
SecurityAdmin:switch>cryptocfg --export -usb KACcert enc_switch1_kac_cert.pem  
Operation succeeded.
```

NOTE

When exporting a certificate to a location other than your home directory, you must specify a fully qualified path that includes the target directory and file name. When exporting to USB storage, certificates are stored by default in a predetermined directory, and you only need to provide a file name for the certificate. An easy way to track exported certificates is by using the base certificate name with the appropriate file extension (***.pem**) and prefixing the name with a character string that identifies the certificate's originator, for example, the switch IP address or host name.

Importing a certificate

1. Log into the switch to which you wish to import the certificate as Admin or SecurityAdmin.
2. Enter the **cryptocfg --import** command with the appropriate parameters.

The following example imports a CP certificate named "enc_switch1_cp_cert.pem" that was previously exported to the external host 192.168.38.245. Certificates are imported to a predetermined directory on the node.

```
SecurityAdmin:swi1th>cryptocfg --import -scp enc_switch1_cp_cert.pem \  
192.168.38.245 mylogin /tmp/certs/enc_switch1_cp_cert.pem  
Password:  
Operation succeeded.
```

The following example imports a CP certificate named "enc_switch1_cp_cert.pem" that was previously exported to USB storage.

```
SecurityAdmin:swi1th>cryptocfg --import -usb enc_switch1_cp_cert.pem \  
enc_switch1_cp_cert.pem  
Operation succeeded.
```


Viewing imported certificates

1. Log into the switch to which you imported the certificates.
2. Enter the **cryptocfg --show -file -all** command to view all imported certificates.

The following example shows the member node CP certificate that was imported earlier to the group leader.

```
SecurityAdmin:switch>cryptocfg --show -file -all  
File name: enc_switch1_cp_cert.pem, size: 1338 bytes
```

NOTE

If the maximum number of certificates is exceeded, the following message is displayed.

Maximum number of certificates exceeded. Delete an unused certificate with the 'cryptocfg -delete -file' command and then try again.

Basic encryption group configuration

An encryption group consists of a set of member nodes that share the same key vault and are managed as a group. At least one node is required to form an encryption group (an encryption group of one would have one member acting as the group leader). An encryption group may include one or more High Availability (HA) clusters and data encryption key (DEK) clusters. An encryption group has the following properties:

- It is identified by a user-defined name.
- It is managed from a designated group leader.
- All group members must share the same key vault.
- When communicating with opaque key vaults, the same master key is used for all encryption operations in the group.
- All encryption engines in a chassis are part of the same encryption group.
- An encryption group may contain up to sixteen encryption engines—up to four nodes with a maximum of four encryption engines per node.

The basic encryption group configuration must be completed before you can set up a key vault or configure a storage device.

Ensure that the following configuration tasks are completed before you create an encryption group:

- [“Management port configuration”](#) on page 87
- [“Encryption switch initialization”](#) on page 90

NOTE

If these configuration steps are not performed, you will not be able to create an HA cluster, perform a first-time encryption, or initiate a re-keying session.

Creating an encryption group

1. Identify one node (a Brocade Encryption Switch or a Brocade DCX or Brocade DCX-4S with an FS8-18 blade) as the designated group leader.
2. Log into the switch as Admin or SecurityAdmin.

3 Basic encryption group configuration

3. Enter the **cryptocfg --create -encgroup** command followed by a name of your choice. The name can be up to 15 characters long, and it can include any alphanumeric characters and underscores. White space or other special characters are not permitted. Successful execution creates an encryption group with the specified name and assigns the role of the group leader to the local node.

The following example creates the encryption group "brocade".

```
SecurityAdmin:switch>cryptocfg --create -encgroup brocade
Encryption group create status: Operation Succeeded.
```

The switch on which you create the encryption group becomes the designated group leader. Once you have created an encryption group, all group-wide configurations, including key vault configuration, adding member nodes, configuring failover policy settings, and setting up storage devices, as well as all encryption management operations, are performed on the group leader.

Setting the key vault type

1. Log into the group leader as Admin or SecurityAdmin.
2. Set the key vault type by entering the **cryptocfg --set -keyvault** command. The options are LKM, RKM, SKM, and NCKA. Successful execution sets the key vault type for the entire encryption group. The following example sets the keyvault type to LKM.

```
SecurityAdmin:switch>cryptocfg --set -keyvault LKM
Set key vault status: Operation Succeeded.
```

Adding a member node to an encryption group

1. Use the **cryptocfg --export -CPcert** command on each node you wish to include in the encryption group and export the CP certificates to an SCP-capable external host or to USB storage. Refer to the section [“Exporting a certificate”](#) on page 93 for instructions.
2. Log into the group leader as Admin or SecurityAdmin.
3. Use the **cryptocfg --import** command to import the CP certificates to the group leader node. You must import the CP certificate of each node you wish to add to the encryption group. Refer to the section [“Importing a certificate”](#) on page 94 for instructions.
4. Enter the **cryptocfg --show -file -all** command on the group leader to verify that you have imported all necessary certificates.
5. On the group leader, register each node you are planning to include in the encryption group. Enter the **cryptocfg --reg -membernode** command with appropriate parameters to register the member node. Specify the member node’s WWN, Certificate filename, and IP address when executing this command. Successful execution of this command distributes all necessary node authentication data to the other members of the group.

```
SecurityAdmin:switch>cryptocfg --reg -membernode \
10:00:00:05:1e:39:14:00 enc_switch1_cert.pem 10.32.244.60
Operation succeeded.
```

NOTE

The order in which member node registration is performed defines group leader succession. At any given time there is only one active group leader in an encryption group. The group leader succession list specifies the order in which group leadership is assumed if the current group leader is not available.

6. Display encryption group member information. This example shows the encryption group "brocade" with two member nodes, one group leader and one regular member. No key vault or HA cluster is configured, and the values for master key IDs are zero.

```
SecurityAdmin:switch>cryptocfg --show -groupmember -all
NODE LIST
Total Number of defined nodes:2
Group Leader Node Name:      10:00:00:05:1e:41:9a:7e
Encryption Group state:     CLUSTER_STATE_CONVERGED

Node Name:                   10:00:00:05:1e:41:9a:7e (current node)
State:                       DEF_NODE_STATE_DISCOVERED
Role:                         GroupLeader
IP Address:                   10.32.244.71
Certificate:                  GL_cpcert.pem
Current Master Key State:     Not configured
Current Master KeyID:        00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00
Alternate Master Key State:  Not configured
Alternate Master KeyID:      00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00

EE Slot: 0
SP state:                     Operational; Need Valid KEK
Current Master KeyID:         00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00
Alternate Master KeyID:      00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00
No HA cluster membership

Node Name:                   10:00:00:05:1e:39:14:00
State:                       DEF_NODE_STATE_DISCOVERED
Role:                         MemberNode
IP Address:                   10.32.244.60
Certificate:                  encl_cpcert.pem
Current Master Key State:     Not configured
Current Master KeyID:        00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00
Alternate Master Key State:  Not configured
Alternate Master KeyID:      00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00

EE Slot:      0
SP state:     Unknown State
Current Master KeyID: 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00
Alternate Master KeyID: 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00
No HA cluster membership
```

Group-wide policy configuration

The group-wide policy parameters as outlined in [Table 7](#) can be set for the entire encryption group on the group leader.

Use the `cryptocfg --set` command with the appropriate parameter to set the values for the policy. Policies are automatically propagated to all member nodes in the encryption group.

TABLE 7 Group-wide policies

Policy name	cryptocfg --set parameters	Description
Failover policy	<code>-failbackmode auto manual</code>	Sets the failback mode. Valid values for failback mode are: <ul style="list-style-type: none"> auto - Enables automatic failback mode. Failback occurs automatically within an HA cluster when an encryption switch or blade that failed earlier has been restored or replaced. Automatic failback mode is enabled by default. manual - Enables manual failback mode. In this mode, failback must be initiated manually when an encryption switch or blade that failed earlier has been restored or replaced.
Heartbeat misses	<code>-hbmisses value</code>	Sets the number of Heartbeat misses allowed in a node that is part of an encryption group before the node is declared unreachable and the Standby takes over. The default <i>value</i> is 3. The range is 1-15 in integer increments only.
Heartbeat timeout	<code>-hbtimeout value</code>	Sets the time-out value for the Heartbeat in seconds. The default <i>value</i> is 2 seconds. Valid <i>values</i> are integers in the range between 1 and 30 seconds. NOTE: The relationship between <code>-hbmisses</code> and <code>-hbtimeout</code> determines the total amount of time allowed before a node is declared unreachable. If a switch does not sense a heartbeat within the heartbeat timeout value, it is counts as a heartbeat miss. The default values result in a total time of 6 seconds (timeout value of two seconds times three misses). A total time of 6 to 10 seconds is recommended. A smaller value may cause a node to be declared unreachable prematurely, while a larger value could result in inefficiency.

Policy Configuration Examples

The following examples illustrate the setting of group-wide policy parameters.

To set the failback mode to manual failback:

```
SecurityAdmin:switch>cryptocfg --set -failbackmode manual
Set failback policy status: Operation Succeeded.
```

To set the Heartbeat misses value to 3:

```
SecurityAdmin:switch>cryptocfg --set -hbmisses 3
Set heartbeat miss status: Operation Succeeded.
```

To set the Heartbeat timeout value to 3 seconds:

```
SecurityAdmin:switch>cryptocfg --set -hbtimeout 3
Set heartbeat timeout status: Operation Succeeded.
```

Key vault configuration

Fabric OS 6.3.0 supports four third-party key management and archival solutions, the NetApp Lifetime Key Management (LKM) appliance, the RSA Key Manager (RKM) appliance, the Hewlett Packard Secure Key Manager (SKM), and the Thales nCipher Key Authority (NCKA). Specific operations must be performed at the key manager to be able to exchange certificates and enable the key vault and the switch to mutually authenticate each other. Refer to [Appendix D, “Supported Key Management Systems”](#) for specific information for each supported key manager.

High Availability (HA) cluster configuration

An HA cluster consists of two encryption engines configured to host the same CryptoTargets and to provide Active/Standby failover and failback capabilities in a single fabric. Failover is automatic (not configurable). Failback occurs automatically by default, but is configurable with a manual failback option. All encryption engines in an HA cluster share the same DEK for a disk or tape LUN.

An HA cluster has the following limitations:

- The encryption engines that are part of an HA cluster must belong to the same encryption group and be part of the same fabric.
- An HA cluster cannot span fabrics and it cannot provide failover/failback capability within a fabric transparent to host MPIO software.

NOTE

Failure to ensure that HA cluster members are part of the same encryption group dissolves the HA cluster and the encryption engines lose their failover capability.

A special kind of HA configuration is a DEK cluster. A DEK cluster consists of a set of HA clusters whose encryption engines can host all paths of the same LUN. Or it can consist of regular standalone encryption engines (not configured into HA clusters) that host paths to the same LUN. All encryption engines in a DEK cluster share the same DEK set, and must be configured, on a per-LUN basis, with the same LUN policies. A DEK cluster works around some of the limitations of an HA cluster. The encryption engines in a DEK cluster may belong to different fabrics and provide failover/failback capability within a SAN by utilizing host MPIO software.

You cannot configure or view a DEK cluster. The DEK cluster is a dynamic grouping established by the Brocade Encryption Switch or FS8-18 blade based on the encryption engines that are configured to host paths to the same LUN.

HA cluster configuration rules

The following rules apply when configuring an HA cluster:

- All HA cluster configuration and related operations must be performed on the group leader.
- I/O sync links must be configured before creating an HA cluster. Refer to the section “[I/O sync link configuration](#)” on page 88 for instructions.
- Configuration changes must be committed before they take effect. Any operation related to an HA cluster that is performed without a commit operation will not survive across switch reboots, power cycles, CP failover, or HA reboots.
- It is recommended that the HA cluster configuration be completed before you configure storage devices for encryption.
- It is mandatory that the two encryption engines in the HA cluster belong to two different nodes for true redundancy. This is always the case for Brocade encryption switches, but is not true if two FS8-18 blades in the same DCX or DCX-4S chassis are configured in the same HA cluster. In Fabric OS version 6.3.0 and later releases, HA cluster creation is blocked when encryption engines belonging to FS8-18 blades in the same DCX or DCX-4S are specified.

Creating an HA cluster

1. Log into the group leader as Admin or SecurityAdmin.
2. Enter the **cryptocfg --create -hacluster** command. Specify a name for the HA cluster and optionally add the node WWN of the encryption engine you wish to include in the HA cluster. Provide a slot number if the encryption engine is a blade. The following example creates an HA cluster named "HAC1" with two encryption engines.

```
SecurityAdmin:switch>cryptocfg --create -hacluster HAC1 \
11:22:33:44:55:66:77:00 10:00:00:05:1e:53:74:87 3
EE Node WWN: 11:22:33:44:55:66:77:00 Slot number: 0 Detected
EE Node WWN: 10:00:00:05:1e:53:74:87 Slot number: 3 Detected
Create HA cluster status: Operation succeeded.
```

3. Enter **cryptocfg --commit** to commit the transaction. Any transaction remains in the "defined" state until it is committed. The commit operation fails if the HA cluster has less than two members.
4. Display the HA cluster configuration by entering the **cryptocfg --show -hacluster -all** command. In the following example, the encryption group "brocade" has one committed HAC1 with two encryption engines.

```
SecurityAdmin:switch>cryptocfg --show -hacluster -all
Encryption Group Name: brocade
Number of HA Clusters: 1

HA cluster name: HAC1 - 1 EE entry
Status:          Committed

      WWN                Slot Number  Status
11:22:33:44:55:66:77:00      0          Online
10:00:00:05:1e:53:74:87      3          Online
```

NOTE

An HA cluster configuration must have two encryption engines before you can commit the transaction with the **cryptocfg --commit** command. To commit an incomplete HA cluster, you have the option to force the commit operation by issuing **cryptocfg --commit -force**. Use the forced commit with caution, because the resulting configuration will not be functional and provide no failover/failback capabilities.

Adding an encryption engine to an HA cluster

1. Log into the group leader as Admin or SecurityAdmin.
2. Enter the **cryptocfg --add -haclustermember** command. Specify the HA cluster name and the encryption engine node WWN. Provide a slot number if the encryption engine is a blade. The following example adds a Brocade FS8-18 in slot 5 to the HA cluster HAC2.

```
SecurityAdmin:switch>cryptocfg --add -haclustermember HAC2 \
10:00:00:60:5b:03:1c:90 5
EE Node WWN: 10:00:00:60:5b:03:1c:90 5 Slot number: 5Detected
Add HA cluster member status: Operation succeeded.
```

3. Add another encryption engine before committing the transaction.

CryptoTarget container configuration

A CryptoTarget container is a configuration of “virtual devices” that is created for each target port hosted on a Brocade Encryption Switch or FS8-18 blade. The container holds the configuration information for a single target, including associated hosts and LUN settings. A CryptoTarget container interfaces between the encryption engine, the external storage devices (targets), and the initiators (hosts) that can access the storage devices through the target ports. Virtual devices redirect the traffic between host and target/LUN to encryption engines so they can perform cryptographic operations.

Virtual targets: Any given physical target port is hosted on one encryption switch or blade. If the target LUN is accessible from multiple target ports, each target port is hosted on a separate encryption switch or blade. There is a one-to-one mapping between virtual target and physical target to the fabric whose LUNs are being enabled for cryptographic operations.

Virtual initiators: For each physical host configured to access a given physical target LUN, a virtual initiator (VI) is generated on the encryption switch or blade that hosts the target port. If a physical host has access to multiple targets hosted on different encryption switches or blades, you must configure one virtual initiator on each encryption switch or blade that is hosting one of the targets. The mapping between physical host and virtual initiator in a fabric is one-to- n , where n is the number of encryption switches or blades that are hosting targets.

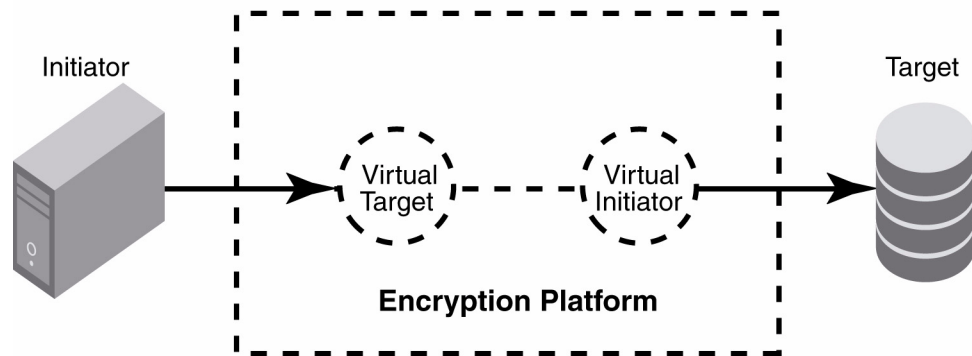


FIGURE 56 Relationship between initiator, virtual target, virtual initiator and target



CAUTION

When configuring a LUN with multiple paths, there is a considerable risk of ending up with potentially catastrophic scenarios where different policies exist for each path of the LUN, or a situation where one path ends up being exposed through the encryption switch and another path has direct access to the device from a host outside the secured realm of the encryption platform. Failure to follow correct configuration procedures for multi-path LUNs results in data corruption. If you are configuring multi-path LUNs as part of an HA cluster or DEK cluster or as a stand-alone LUN accessed by multiple hosts, follow the instructions described in the section [“Configuring a multi-path Crypto LUN”](#) on page 117.

Gathering information

Before you begin, have the following information ready:

- The switch WWNs of all nodes in the encryption group. Use the **cryptocfg --show -groupmember -all** command to gather this information.
- The port WWNs of the targets whose LUNs are being enabled for data-at-rest encryption.
- The port WWNs of the hosts (initiators) which should gain access to the LUNs hosted on the targets.

Any given target may have multiple ports through which a given LUN is accessible and the ports are connected to different fabrics for redundancy purposes. Any given target port through which the LUNs are accessible must be hosted on only one Encryption switch (or pair in case of HA deployment). Another such target port should be hosted on a different encryption switch either in the same fabric or in a different fabric based on host MPIO configuration.

A given host port through which the LUNs are accessible is hosted on the same encryption switch on which the target port (CryptoTarget container) of the LUNs is hosted.

NOTE

It is recommended you complete the encryption group and HA cluster configuration before configuring the CryptoTarget containers.

Frame redirection

Name Server-based frame redirection enables the Brocade encryption switch or blade to be deployed transparently to hosts and Targets in the fabric.

NS-based frame redirection is enabled as follows:

- You first create a zone that includes host (H) and target (T). This may cause temporary traffic disruption to the host.
- You then create a CryptoTarget container for the target and configure the container to allow access to the initiator.
- When you commit the transaction, a special zone called a “redirection zone” is generated automatically. The redirection zone includes the host (H), the virtual target (VT), the virtual initiator (VI), and the target (T).
- When configuring multi-path LUNs do not commit the CryptoTarget container configuration before you have performed the following steps in sequence to prevent data corruption. Refer to the section [“Configuring a multi-path Crypto LUN”](#) on page 117 for more information.
 - Complete all zoning for ALL hosts that should gain access to the targets.
 - Complete the CryptoTarget container configuration for ALL target ports in sequence, including adding the hosts that should gain access to these targets.

Host-target zoning must precede any CryptoTarget configuration.

NOTE

To enable frame redirection, the host and target edge switches must run Fabric OS v6.1.1 and Fabric OS v5.3.1.b or later firmware to ensure host and target connectivity with legacy platforms. In McDATA fabrics, the hosts and the switches hosting the targets require firmware versions M-EOSc 9.8 and M-EOSn 9.8. Only the M6140, M4700F, McDATA 4400, and the Brocade Intrepid 10000 support frame redirection. Refer to Appendix E, [Table 25](#) on page 201 for more information.

Creating an initiator - target zone

1. Log into the group leader as Admin or FabricAdmin.
2. Determine the initiator PWWN. Enter the **nsshow** command to view the devices connected to this switch. In the following example, the port name 10:00:00:00:c9:2b:c9:3a is the initiator PWWN.

```
FabricAdmin:switch>nsshow
{
  Type Pid   COS PortName                               NodeName                               TTL(sec)
  N 010600; 2,3;10:00:00:00:c9:2b:c9:3a;20:00:00:00:c9:2b:c9:3a; na
  NodeSymb: [35] "Emulex LP9002 FV3.82A1 DV5-4.81A4 "
  Fabric Port Name: 20:06:00:05:1e:41:9a:7e
  Permanent Port Name: 10:00:00:00:c9:2b:c9:3a
  Port Index: 6
  Share Area: No
  Device Shared in Other AD: No
  Redirect: No
  The Local Name Server has 1 entry }
```

The **nsshow** command shows all devices on the switch, and the output can be lengthy. To retrieve only the initiator PWWN, do a pattern search of the output based on the initiator Port ID (a hex number). In the following example, The PID is 010600, where 01 indicates the domain and 06 the port number.

```
FabricAdmin:switch>nsshow | grep 0106
N 010600; 2,3;10:00:00:00:c9:2b:c9:3a;20:00:00:00:c9:2b:c9:3a; na
```

3. Determine the target PWWN. Enter the **nsscanshow** command to review the remote switch information. In the following example, the port name 20:0c:00:06:2b:0f:72:6d is the target PWWN.

```
FabricAdmin:switch>nsscanshow
nsscanshow for remote switches:
Switch entry for 2
state rev owner
known v611 0xffffc01
Device list: count 13
Type Pid COS   PortName                               NodeName
NL  0208d3; 3;20:0c:00:06:2b:0f:72:6d;20:00:00:06:2b:0f:72:6d;
FC4s: FCP
PortSymb: [55] "LSI7404XP-LC BR A.1 03-01081-02D FW:01.03.06 Port 1 "
Fabric Port Name: 20:08:00:05:1e:34:e0:6b
Permanent Port Name: 20:0c:00:06:2b:0f:72:6d
Port Index: 8
Share Area: No
Device Shared in Other AD: No
Redirect: No
```

Alternately use **nsscanshow | grep target PID** to obtain the target PWWN only.

```
FabricAdmin:switch>nsscanshow | grep 0208
NL  0208d3; 3;20:0c:00:06:2b:0f:72:6d;20:00:00:06:2b:0f:72:6d;
```

4. Create a zone that includes the initiator and a LUN target. Enter the **zonecreate** command followed by a zone name, the initiator PWWN and the target PWWN.

```
FabricAdmin:switch>zonecreate itzone, "10:00:00:00:c9:2b:c9:3a; \
20:0c:00:06:2b:0f:72:6d"
```

5. Create a zone configuration that includes the zone you created in step 4. Enter the **cfgcreate** command followed by a configuration name and the zone member name.

```
FabricAdmin:switch>cfgcreate itcfg, itzone
```

6. Enable the zone configuration.

```
FabricAdmin:switch>cfgenable itcfg
You are about to enable a new zoning configuration.
This action will replace the old zoning configuration with the
current configuration selected.
Do you want to enable 'itcfg' configuration (yes, y, no, n): [no] y
zone config"itcfg" is in effect
Updating flash ...
```

Creating a CryptoTarget container

1. Log into the group leader as Admin or FabricAdmin.
2. Enter the **cryptocfg --create -container** command. Specify the type of the container, (disk or tape), followed by a name for the CryptoTarget container, the encryption engine's node WWN, and the target's Port WWN and node WWN. Provide a slot number if the encryption engine is a blade.
 - The CryptoTarget container name can be up to 31 characters in length and may include any alphanumeric characters, hyphens, and underscore characters.
 - You may add initiators at this point or after you create the container.

The following example creates a disk container named `my_disk_tgt1`. The initiator is added in step 3.

```
FabricAdmin:switch>cryptocfg --create -container disk my_disk_tgt \
10:00:00:00:05:1e:41:9a:7e 20:0c:00:06:2b:0f:72:6d 20:00:00:06:2b:0f:72:6d
Operation Succeeded
```

3. Add an initiator to the CryptoTarget container. Enter the **cryptocfg --add -initiator** command followed by the initiator port WWN and the node WWN.

Note that the initiator port WWN must also be added to the LUN when the LUN is added to the CryptoTarget container.

```
FabricAdmin:switch>cryptocfg --add -initiator my_disk_tgt \
10:00:00:00:c9:2b:c9:3a 20:00:00:00:c9:2b:c9:3a
Operation Succeeded
```

4. Commit the transaction. The commit operation creates the virtual devices and the redirection zone that routes traffic through these devices.

```
FabricAdmin:switch>cryptocfg --commit
Operation Succeeded
```



CAUTION

When configuring a multi-path LUN, you must complete the CryptoTarget container configuration for ALL target ports in sequence and add the hosts that should gain access to these ports before committing the container configuration. Failure to do so results in data corruption. Refer to the section [“Configuring a multi-path Crypto LUN”](#) on page 117 for specific instructions.

5. Display the CryptoTarget container configuration. The virtual initiator and virtual target have been created automatically upon commit, and there are no LUNs configured yet.

```
FabricAdmin:switch>cryptocfg --show -container my_disk_tgt -cfg
Container name:      my_disk_tgt
Type:               disk
EE node:           10:00:00:05:1e:41:9a:7e
EE slot:           0
Target:            20:0c:00:06:2b:0f:72:6d 20:00:00:06:2b:0f:72:6d
VT:                20:00:00:05:1e:41:4e:1d 20:01:00:05:1e:41:4e:1d
Number of host(s):  1
Configuration status: committed
Host:              10:00:00:00:c9:2b:c9:3a 20:00:00:00:c9:2b:c9:3a
VI:                20:02:00:05:1e:41:4e:1d 20:03:00:05:1e:41:4e:1d
Number of LUN(s):  0
Operation Succeeded
```

6. Display the redirection zone. It includes the host, the target, the virtual initiator, and the virtual target.

```
FabricAdmin:switch>cfgshow
Defined configuration:
cfg:  itcfg          itzone
cfg:  r_e_d_i_r_c_fg
      red_1109_brcd200c00062b0f726d200200051e414e1d; red_____base
zone: itzone 10:00:00:00:c9:2b:c9:3a; 20:0c:00:06:2b:0f:72:6d
zone: red_1109_brcd200c00062b0f726d200200051e414e1d
      10:00:00:00:c9:2b:c9:3a; 20:0c:00:06:2b:0f:72:6d;
      20:02:00:05:1e:41:4e:1d; 20:00:00:05:1e:41:4e:1d
zone: red_____base
      00:00:00:00:00:00:00:01; 00:00:00:00:00:00:00:02;
      00:00:00:00:00:00:00:03; 00:00:00:00:00:00:00:04
Effective configuration:
cfg:  itcfg
zone: itzone 10:00:00:00:c9:2b:c9:3a
      20:0c:00:06:2b:0f:72:6d
```

NOTE

You may view the frame redirection zone with the `cfgshow` command, but you cannot use the zone for any other applications that use frame redirection. Do not perform any further operations with this zone, such as deleting the zone, or adding the zone to a different configuration. Such operations may result in disruptive behavior including data corruption on the LUN.

Removing an initiator from a CryptoTarget container

You may remove one or more initiators from a given CryptoTarget container. This operation removes the initiators' access to the target port.

If the initiator has access to multiple targets and you wish to remove access to all targets, follow the procedure described to remove the initiator from every CryptoTarget container that is configured with this initiator.

NOTE

Stop all traffic between the initiator you intend to remove and its respective target ports. Failure to do so results in I/O failure between the initiator and the target port.

1. Log into the group leader as Admin or FabricAdmin.
2. Enter the **cryptocfg --remove -initiator** command. Specify the CryptoTarget container name followed by one or more initiator port WWNs. The following example removes one initiator from the CryptoTarget container "my_disk_tgt".

```
FabricAdmin:switch>cryptocfg --rem -initiator my_disk_tgt
10:00:00:00:c9:2b:c9:3a
Operation Succeeded
```

3. Commit the transaction.

```
FabricAdmin:switch>cryptocfg --commit
Operation Succeeded
```



CAUTION

When configuring a multi-path LUN, you must remove all initiators from all CryptoTarget containers in sequence before committing the transaction. Failure to do so may result in a potentially catastrophic situation where one path ends up being exposed through the encryption switch and another path has direct access to the device from a host outside the protected realm of the encryption platform. Refer to the section [“Configuring a multi-path Crypto LUN”](#) on page 117 for more information.

Deleting a CryptoTarget container

You may delete a CryptoTarget container to remove the target port from a given encryption switch or blade. Deleting a CryptoTarget container removes the virtual target and all associated LUNs from the fabric.

Before deleting a container, be aware of the following:

- Stop all traffic to the target port for which the CryptoTarget container is being deleted. Failure to do so will cause data corruption (a mix of encrypted data and cleartext data will be written to the LUN).
- Deleting a CryptoTarget container while a re-key or first-time encryption session causes all data to be lost on the LUNs that are being re-keyed. Ensure that no re-key or first time encryption sessions are in progress before deleting a container. Use the **cryptocfg --show -rekey -all** command to determine the runtime status of the session. If for some reason, you need to delete a container while re-keying, when you create a new container, be sure the LUNs added to the container are set to **cleartext**. You can then start a new re-key session on clear text LUNs.

3 CryptoTarget container configuration

1. Log into the group leader as Admin or FabricAdmin.
2. Enter the **cryptocfg --delete -container** command followed by the CryptoTarget container name. The following example removes the CryptoTarget container “my_disk_tgt”.

```
FabricAdmin:switch>cryptocfg --delete -container my_disk_tgt  
Operation Succeeded
```

3. Commit the transaction.

```
FabricAdmin:switch>cryptocfg --commit  
Operation Succeeded
```



CAUTION

When configuring a multi-path LUN, you must remove all necessary CryptoTarget containers in sequence before committing the transaction. Failure to do so may result in a potentially catastrophic situation where one path ends up being exposed through the encryption switch and another path has direct access to the device from a host outside the protected realm of the encryption platform. Refer to the section “[Configuring a multi-path Crypto LUN](#)” on page 117 for more information.

Moving a CryptoTarget container

You can move a CryptoTarget container from one encryption engine to another. The encryption engines must be part of the same fabric and the same encryption group, and the encryption engines must be online for this operation to succeed. The two encryption engines do not need to be part of the same HA cluster. This operation permanently transfers the encryption engine association of a given CryptoTarget container from an existing encryption engine to an alternate encryption engine.

NOTE

If a CryptoTarget container is moved in a configuration involving FCR, the LSAN zones and manually created redirect zones will need to be reconfigured with new VI and VT WWNs. Refer to the section “[Deployment in Fibre Channel routed fabrics](#)” on page 139 for instructions on configuring encryption in an FCR deployment scenario.

1. Log into the group leader as Admin or FabricAdmin.
2. Enter the **cryptocfg --move -container** command followed by the CryptoTarget container name and the node WWN of the encryption engine to which you are moving the CryptoTarget container. Provide a slot number if the encryption engine is a blade.

```
FabricAdmin:switch>cryptocfg --move -container my_disk_tgt \  
10:00:00:05:1e:53:4c:91  
Operation Succeeded
```

3. Commit the transaction.

```
FabricAdmin:switch>cryptocfg --commit  
Operation Succeeded
```

Crypto LUN configuration

A Crypto LUN is the LUN of a target disk or tape storage device that is enabled for and capable of data-at-rest encryption. Crypto LUN configuration is done on a per-LUN basis. You configure the LUN for encryption by explicitly adding the LUN to the CryptoTarget container and turning on the encryption property and policies on the LUN. Any LUN of a given target that is not enabled for encryption must still be added to the CryptoTarget container with the **cleartext** policy option.

- The general procedures described in this section apply to both disk and tape LUNs. The specific configuration procedures differ with regard to encryption policy and parameter setting.
- You configure the Crypto LUN on the group leader. You need the FabricAdmin role to perform LUN configuration tasks.
- Only one path for a LUN per encryption engine and only one path for a LUN per HA cluster pair is supported. When an actual LUN has multiple paths, each path must be hosted on a separate encryption engine or HA cluster pair as a Crypto Target Container (CTC). This applies to both the active path and passive path. Never host both an active path and passive path to a LUN on the same encryption engine or HA cluster pair.



CAUTION

When configuring a LUN with multiple paths (which means the LUN is exposed and configured on multiple Crypto Target containers located on the same Encryption switch or blade or on different encryption switches or blades), the same LUN policies must be configured on all of the LUN's paths. Failure to configure all LUN paths with the same LUN policies results in data corruption. If you are configuring multi-path LUNs as part of a HA cluster or DEK cluster or as a stand-alone LUN accessed by multiple hosts, follow the instructions described in the section [“Configuring a multi-path Crypto LUN”](#) on page 117.

Discovering a LUN

When adding a LUN to a CryptoTarget container, you must specify a LUN Number. The LUN Number needed for configuring a given Crypto LUN is the LUN Number as exposed to a particular initiator.

The Brocade Encryption platform provides LUN discovery services through which you can identify the exposed LUN number for a specified initiator. If you already know the exposed LUN numbers for the various initiators accessing the LUN, you may skip the LUN discovery step and directly configure the Crypto LUN.

1. Log into the group leader as Admin or FabricAdmin.
2. Enter the **cryptocfg --discoverLUN** command followed by the CryptoTarget container Name.

```
FabricAdmin:switch>cryptocfg --discoverLUN my_disk_tgt
Container name: my_disk_tgt
Number of LUN(s): 1
Host: 10:00:00:00:c9:2b:c9:3a
LUN number: 0x0
LUN serial number: 200000062B0F726D0C000000
Key ID state: Key ID not available
Key ID: 3a:21:6a:bd:f2:37:d7:ea:6b:73:f6:19:72:89:c6:4f
```



CAUTION

When configuring a LUN with multiple paths, perform the LUN discovery on each of the Crypto Target containers for each of the paths accessing the LUN and verify that the serial number for these LUNs discovered from these Crypto Target containers are the same. This indicates and validates that these Crypto Target containers are indeed paths to the same LUN. Refer to the section [“Configuring a multi-path Crypto LUN”](#) on page 117 for more information.

Configuring a Crypto LUN

You configure a Crypto LUN by adding the LUN to the CryptoTarget container and enabling the encryption property on the Crypto LUN. The LUNs of the target which are not enabled for encryption must still be added to the CryptoTarget container with the **cleartext** policy option.

You can add a single LUN to a CryptoTarget container, or you can add multiple LUNs by providing a range of LUN Numbers. When adding a single LUN, you can either provide a 16-bit (2 byte) hex value of the LUN Number, for example, 0x07. Alternately you can provide a 64-bit (8 byte) value in WWN or LUN ID format, for example, 00:07:00:00:00:00:00:00. When adding a range of LUN Numbers, you may use two byte hex values or decimal numbers.

NOTE

LUN configurations and modifications must be committed to take effect. There is an upper limit of 25 on the number of LUNs you can add or modify in a single commit operation. Attempts to commit a configuration that exceeds this maximum will fail. Note that there is also a five second delay before the commit operation takes effect. In addition to the above limit of 25 per commit, make sure the LUNs in previously committed LUN configurations and LUN modifications have a LUN state of **Encryption Enabled** before creating and committing another batch of 25 LUN configurations or LUN modifications.

The device type (disk or tape) is set at the CryptoTarget container level. You cannot add a tape LUN to a CryptoTarget container of type “disk” and vice versa.

It is recommended that you configure the LUN state and encryption policies at this time. You can add these settings later with the **cryptocfg --modify -LUN** command, but not all options are modifiable. Refer to the section [“Crypto LUN parameters and policies”](#) on page 112 for LUN configuration parameters. Refer to the section [“Creating a tape pool”](#) on page 123 for tape pool policy parameters.

NOTE

If you are using VMware virtualization software or any other configuration that involves mounted file systems on the LUN, you must enable first-time encryption at the time when you create the LUN by setting the **--enable_encexistingdata** option with the **--add -LUN** command. Failure to do so permanently disconnects the LUN from the host and causes data to be lost and unrecoverable.

Log into the group leader as Admin or FabricAdmin.

3. Enter the **cryptocfg --add -LUN** command followed by the CryptoTarget container Name, the LUN number or a range of LUN numbers, the PWWN and NWWN of the initiators that should be able to access the LUN. If you are using Datafort encryption format, you can use the **-encryption_format** option to set the format to **DF_compatible** (the default is **Native**). The following example adds a disk LUN enabled for encryption.

```
FabricAdmin:switch>cryptocfg --add -LUN my_disk_tgt 0x0 \
10:00:00:00:c9:2b:c9:3a 20:00:00:00:c9:2b:c9:3a -encrypt
Operation Succeeded
```

4. Commit the configuration.

```
FabricAdmin:switch>cryptocfg --commit
Operation Succeeded
```



CAUTION

When configuring a LUN with multiple paths, do not commit the configuration before you have added all the LUNs with identical policy settings and in sequence to each of the Crypto Target containers for each of the paths accessing the LUNs. Failure to do so results in data corruption. Refer to the section [“Configuring a multi-path Crypto LUN”](#) on page 117.

5. Display the LUN configuration. The following example shows default values.

```
FabricAdmin:switch>cryptocfg --show -LUN my_disk_tgt0 \
10:00:00:00:c9:2b:c9:3a -cfg
EE node: 10:00:00:05:1e:41:9a:7e
EE slot: 0
Target: 20:0c:00:06:2b:0f:72:6d 20:00:00:06:2b:0f:72:6d
VT: 20:00:00:05:1e:41:4e:1d 20:01:00:05:1e:41:4e:1d
Number of host(s): 1
Configuration status: committed
Host: 10:00:00:00:c9:2b:c9:3a 20:00:00:00:c9:2b:c9:3a
VI: 20:02:00:05:1e:41:4e:1d 20:03:00:05:1e:41:4e:1d
LUN number: 0x0
LUN type: disk
LUN status: 0
Encryption mode: encrypt
Encryption format: native
Encrypt existing data: enabled
Rekey: disabled
Key ID: not available
Operation Succeeded
```

Removing a LUN from a CryptoTarget container

You can remove a LUN from a given CryptoTarget container if it is no longer needed. Stop all traffic I/O from the initiators accessing the LUN before removing the LUN to avoid I/O failure between the initiators and the LUN. If the LUN is exposed to more than one initiator under different LUN Numbers, remove all exposed LUN Numbers.

1. Log into the group leader as Admin or FabricAdmin.
2. Enter the **cryptocfg --remove -LUN** command followed by the CryptoTarget container name, the LUN Number, and the initiator PWWN.

3 Crypto LUN configuration

```
FabricAdmin:switch>cryptocfg --remove -LUN my_disk_tgt 0x0  
10:00:00:00:c9:2b:c9:3a  
Operation Succeeded
```

3. Commit the configuration with the **-force** option to completely remove the LUN and all associated configuration data in the configuration database. The data remains on the removed LUN in an encrypted state.

```
FabricAdmin:switch>cryptocfg --commit -force  
Operation Succeeded
```



CAUTION

In case of multiple paths for a LUN, each path is exposed as a CryptoTarget container in the same encryption switch or blade or on different encryption switches or blades within the encryption group. In this scenario you must remove the LUNs from all exposed CryptoTarget containers before you commit the transaction. Failure to do so may result in a potentially catastrophic situation where one path ends up being exposed through the encryption switch and another path has direct access to the device from a host outside the protected realm of the encryption platform. Refer to the section “[Configuring a multi-path Crypto LUN](#)” on page 117 for more information.

Crypto LUN parameters and policies

[Table 8](#) shows the encryption parameters and policies that can be specified for a disk or tape LUN, during LUN configuration (with the **cryptocfg --add LUN** command). Some policies are applicable only to disk LUNs, and some policies are applicable only to tape LUNs. It is recommended that you plan to configure all the LUN state and encryption policies with the **cryptocfg --add LUN** command. You can use the **cryptocfg --modify -LUN** command to change some of the settings, but not all options are modifiable.

NOTE

LUN policies are configured at the LUN-level but apply to the entire HA or DEK cluster. For multi-path LUNs exposed through multiple target ports and thus configured on multiple Crypto Target containers on different encryption engines in an HA cluster or DEK cluster, the same LUN policies must be configured. Failure to do so results in unexpected behavior and may lead to data corruption.

The tape policies specified at the LUN configuration level take effect if you do not create tape pools or configure policies at the tape pool level.

TABLE 8 LUN parameters and policies

Policy name	Command parameters	Description
LUN state Disk LUN: yes Tape LUN: No Modify? No	-lunstate encrypted cleartext	Sets the Encryption state for the LUN. Valid values are: <ul style="list-style-type: none"> cleartext - Default LUN state. Refer to policy configuration considerations for compatibility with other policy settings. encrypted - Metadata on the LUN containing the key ID of the DEK that was used for encrypting the LUN is used to retrieve the DEK from the key vault. DEKs are used for encrypting and decrypting the LUN.
Key ID Disk LUN: yes Tape LUN: No Modify? No	-keyID <i>Key_ID</i>	Specifies the key ID. Use this option <i>only</i> if the LUN was encrypted but does not include the metadata containing the key ID for the LUN. This is a rare case for LUNs encrypted in Native (Brocade) mode. However for LUNs encrypted with DataFort v2.0, a key ID is required, because these LUNs do not contain any metadata.
Encryption format Disk LUN: yes Tape LUN: yes Modify? Yes	-encryption_format native DF_compatible	Sets the encryption format. Valid values are: <ul style="list-style-type: none"> Native - The LUN is encrypted or decrypted using the Brocade encryption format (metadata format and algorithm). This is the default setting. DF_compatible - The LUN is encrypted or decrypted using the NetApp DataFort encryption format (metadata format and algorithm). Use of this format requires a NetApp DataFort-compatible license. <p>NOTE: On tapes written in DataFort format, the encryption switch or blade cannot read and decrypt files with a block size of one MB or greater.</p>
Encryption policy Disk LUN: yes Tape LUN: Yes Modify? Yes	-encrypt -cleartext	Enables or disables a LUN for encryption. Valid values are: <ul style="list-style-type: none"> cleartext - Encryption is disabled. This is the default setting. When the LUN policy is set to cleartext the following policy parameters are invalid and generate errors when executed: -enable_encexistingdata, -enable_rekey, and -key_lifespan. When a LUN is added in DataFort-compatible encryption format, cleartext is not a valid policy option. encrypt - The LUN is enabled to perform encryption.
Existing data encryption Disk LUN: yes Tape LUN: No Modify? Yes	-enable_encexistingdata -disable_encexistingdata	Specifies whether or not existing data on the LUN should be encrypted. By default, encryption of existing data is disabled. Encryption policy must be set to -enable_encexistingdata , and the LUN state must be set to cleartext (default). If the encryption policy is cleartext , the existing data on the LUN will be overwritten.
Re-key policy Disk LUN: yes Tape LUN: No Modify? Yes	-enable_rekey <i>time_period</i> < <i>days</i> > -disable_rekey	Enables or disables the auto re-keying feature on a specified disk LUN. This policy is not valid for tape LUNs. By Default, the automatic re-key feature is disabled. Enabling automatic re-keying is valid only if the LUN policy is set to -encrypt . You must specify a time period in days when enabling Auto Re-keying to indicate the interval at which automatic re-keying should take place.
Key lifespan Disk LUN: No Tape LUN: Yes Modify? Disks only. Tape: No	-key_lifespan <i>time_in_days</i> none	Specifies the life span of the encryption key in days. The key will expire after the specified number of days. Accepted values are integers from 1 to 2982616. The default value is none, which means the key does not expire. On tape LUNs, the key life span cannot be modified after it is set.

Modifying Crypto LUN parameters

You can modify one or more policies of an existing Crypto LUN with the `cryptocfg --modify -LUN` command. If the modification applies to multiple LUNs, you may specify a LUN number range.

NOTE

A maximum of 25 LUNs can be added or modified in a single commit operation. Attempts to commit configurations or modifications that exceed this maximum fail with a warning. Note that there is a five second delay before the commit operation takes effect. Make sure the LUNs in previously committed LUN configurations and LUN modifications have a LUN state of **Encryption Enabled** before creating and committing another batch of 25 LUN configurations or LUN modifications.

The following example disables automatic re-keying operations on the disk LUN “my_disk_tgt.”

1. Log into the group leader as Admin or FabricAdmin.
2. Enter the `cryptocfg --modify -LUN` command followed by the CryptoTarget container name, the LUN Number, the initiator PWWN, and the parameter you wish to modify.

```
FabricAdmin:switch>cryptocfg --modify -LUN my_disk_tgt 0x0
10:00:00:00:c9:2b:c9:3a -disable_rekey
Operation Succeeded
```

3. Commit the configuration.

```
FabricAdmin:switch>cryptocfg --commit
Operation Succeeded
```



CAUTION

When configuring a LUN with multiple paths, do not commit the configuration before you have modified all the LUNs with identical policy settings and in sequence for each of the Crypto Target containers for each of the paths accessing the LUNs. Failure to do so results in data corruption. Refer to the section [“Configuring a multi-path Crypto LUN”](#) on page 117.

LUN modification considerations

Make sure you understand the ramifications of modifying LUN policy parameters (such as encrypt/cleartext) for LUNs that are online and already being utilized. The following restrictions apply when modifying LUN policy parameters for disk LUNs:

- When you change LUN policy from **encrypt** to **cleartext**, you will wipe out all encrypted data stored on the LUN the next time data is written to that LUN. The following policy parameters are disabled: **-enable_encexistingdata**, **-enable_rekey**.
- When you change the LUN policy back to **encrypt**, for example, by force-enabling the LUN, **-enable_encexistingdata** and **-enable_rekey** are disabled by default, and you must configure both options again.
- When you add a LUN as **cleartext** and later you want to change the LUN policy from **cleartext** to **encrypt**, you must set the **-enable_encexistingdata** option. If you do not, all data on that LUN is lost, and cannot be recovered.

For tape LUNs **-enable_encexistingdata** and **-enable_rekey** are not valid and therefore cannot be modified. The **-key_lifespan** parameter is valid for tape LUNs but it cannot be modified after it is set. When you attempt to execute these parameters while modifying a tape LUN, the system returns an error.

For specific handling of encryption policy changes when using DF-compatible encryption format, refer to Appendix D “[DF-compatibility support for disk LUNs](#)” on page 195 and “[DF-compatibility support for tape LUNs](#)” on page 199.

Force-enabling a disabled disk LUN for encryption

You can force a disk LUN to become enabled for encryption when encryption is disabled on the LUN. A LUN may become disabled for various reasons, such as a change in policy from **encrypt** to **cleartext** when encrypted data (and metadata) exist on the LUN, a conflict between LUN policy and LUN state, or a missing DEK in the key vault. Force-enabling a LUN while metadata exist on the LUN may result in a loss of data and should be exercised with caution. Refer to Chapter 6, “[LUN policy troubleshooting](#)” on page 185 for a description of conditions under which a LUN may be disabled, and for recommendations on re-enabling the LUN while minimizing the risk of data loss.

This procedure must be performed on the local switch that is hosting the LUN. No commit is required to force-enable after executing this command.

1. Log into the switch that hosts the LUN as Admin or FabricAdmin.
2. Enter the **cryptocfg --enable -LUN** command followed by the CryptoTarget container name, the LUN Number, and the initiator PWWN.

```
FabricAdmin:switch>cryptocfg --enable -LUN my_disk_tgt 0x0 \
10:00:00:00:c9:2b:c9:3a
Operation Succeeded
```

Configuring a tape LUN

This example shows how to configure a tape storage device. The basic setup procedure is the same as for disk devices. Only a subset of configuration options and policy settings are available for tape LUNs. Refer to [Table 8](#) on page 113 for tape LUN configuration options.

1. Create a zone that includes the initiator (host) and the target port. Refer to the section “[Creating an initiator - target zone](#)” on page 104 for instructions.
2. Create a CryptoTarget container of type **tape**. Refer to the section “[Creating a CryptoTarget container](#)” on page 105 for instructions.
 - a. Create the container, allowing the encryption format to default to Native.

```
FabricAdmin:switch>cryptocfg --create -container tape my_tape_tgt \
10:00:00:05:1e:41:9a:7e 20:0c:00:06:2b:0f:72:6d 20:00:00:06:2b:0f:72:6d
Operation Succeeded
```

- b. Add an initiator to the CryptoTarget container “my_tape_tgt”.

```
FabricAdmin:switch>cryptocfg --add -initiator my_tape_tgt \
10:00:00:00:c9:2b:c9:3a 20:00:00:00:c9:2b:c9:3a
Operation Succeeded
```

- c. Commit the transaction.

```
FabricAdmin:switch>cryptocfg --commit
Operation Succeeded
```

3. Configure the Crypto tape LUN. Refer to the section “[Configuring a Crypto LUN](#)” on page 110 for instructions.

3 Crypto LUN configuration

- a. Discover the LUN.

```
FabricAdmin:switch>cryptocfg --discoverLUN my_tape_tgt
Container name:      my_tape_tgt
Number of LUN(s):   1
Host:               10:00:00:00:c9:2b:c9:3a
LUN number:         0x0
LUN serial number:
Key ID state:       Key ID not Applicable
```

- b. Add the LUN to the tape CryptoTarget container. The following example enables the LUN for encryption. There is a maximum of eight tape LUNs per Initiator in a container.

```
FabricAdmin:switch>cryptocfg --add -LUN my_tape_tgt 0x0 \
10:00:00:00:c9:2b:c9:3a 20:00:00:00:c9:2b:c9:3a -encrypt
Operation Succeeded
```

NOTE

When changing the tape LUN policy from **encrypt** to **cleartext** or from **cleartext** to **encrypt**, or the encryption format from Brocade **native** to **DF-compatible** while data is being written to or read from a tape backup device, the policy change is not enforced until the current process completes and the tape is unmounted, rewound, or overwritten. Refer to the section [“Impact of tape LUN configuration changes”](#) on page 124 for more information.

- c. Commit the configuration.

```
FabricAdmin:switch>cryptocfg --commit
Operation Succeeded
```

- d. Display the LUN configuration

```
FabricAdmin:switch>cryptocfg --show -LUN my_tape_tgt 0x0 \
10:00:00:00:c9:2b:c9:3a -cfg
EE node:           10:00:00:05:1e:41:9a:7e
EE slot:           0
Target:           20:0c:00:06:2b:0f:72:6d 20:00:00:06:2b:0f:72:6d
VT:              20:00:00:05:1e:41:4e:1d 20:01:00:05:1e:41:4e:1d
Number of host(s): 1
Configuration status: committed
Host:            21:00:00:e0:8b:89:9c:d5 20:00:00:e0:8b:89:9c:d5
VI:             10:00:00:00:c9:2b:c9:3a 20:03:00:05:1e:41:4e:31
LUN number:      0x0
LUN type:        tape
LUN status:      0
Encryption mode: encrypt
Encryption format: DF_compatible
Tape type:       tape
Key life:        90 (day)
Volume/Pool label:
Operation succeeded.
```

Modify example

The following is an example of the use of the **cryptocfg -- modify** command. This example changes the encryption format from Brocade native to DF-compatible.

```
FabricAdmin:switch>cryptocfg --modify -LUN my_tape_tgt 0x0
10:00:00:00:c9:2b:c9:3a -encryption-format DF_compatible
Operation Succeeded
```

Configuring a multi-path Crypto LUN

A single LUN may be accessed over multiple paths. A multi-path LUN is exposed and configured on multiple CryptoTarget Containers located on the same encryption switch or blade or on different encryption switches or blades.



CAUTION

When configuring a LUN with multiple paths, there is a considerable risk of ending up with potentially catastrophic scenarios where different policies exist for each path of the LUN, or a situation where one path ends up being exposed through the encryption switch and other path has direct access to the device from a host outside the secured realm of the encryption platform. Failure to follow proper configuration procedures for multi-path LUNs results in data corruption.

To avoid the risk of data corruption, it is of utmost importance that you observe the following rules when configuring multi-path LUNs:

- During the initiator-target zoning phase, complete in sequence all zoning for ALL hosts that should gain access to the targets before committing the zoning configuration.
- Complete the CryptoTarget container configuration for ALL target ports in sequence and add the hosts that should gain access to these ports *before* committing the container configuration. Upon commit, the hosts lose access to all LUNs until the LUNs are explicitly added to the Crypto Target containers.
- When configuring the LUNs, the *same* LUN policies must be configured for ALL paths of ALL LUNs. Failure to configure all LUN paths with the same LUN policies results in data corruption.

Multi-path LUN configuration example

Figure 57 on page 120 shows a single LUN on a dual-port target that is accessed over two paths by a dual-port host. The two encryption switches form an encryption group and an HA cluster. The following example illustrates a simplified version of a multi-path LUN configuration.

1. Create zoning between host port 1 and target port 1. Refer to the section “[Creating an initiator - target zone](#)” on page 104 for instructions.
2. Create zoning between host port 2 and target port 2. Refer to the section “[Creating an initiator - target zone](#)” on page 104 for instructions.
3. On the group leader encryption switch (switch 1), create a CryptoTarget container for each target port and add the hosts in sequence. Do NOT commit the configuration until you have created all CryptoTarget containers and added all hosts to the respective containers.
 - a. Create a CryptoTarget container (CTC1) for target port 1 to be hosted on the encryption engine of encryption switch 1. Refer to the section “[Creating a CryptoTarget container](#)” on page 105 for instructions on steps b. through e.

```
FabricAdmin:switch>cryptocfg --create -container disk CTC1 \  
<switch 1 WWN> 0 <Target Port 1 WWN> <Target NWWN>
```

- b. Create a CryptoTarget container (CTC2) for target port 2 to be hosted on the encryption engine of encryption switch 2.

```
FabricAdmin:switch>cryptocfg --create -container disk CTC2 \  
<switch 2 WWN> 0 <Target Port2 WWN> <Target NWWN>
```

3 Configuring a multi-path Crypto LUN

- c. Add host port 1 to the container CTC1.

```
FabricAdmin:switch>cryptocfg --add -initiator <CTC1> <Host Port1 WWN> \  
<Host NWWN>
```

- d. Add host port 2 to the container CTC2.

```
FabricAdmin:switch>cryptocfg --add -initiator <CTC2> <Host Port2 WWN> \  
<Host NWWN>
```

- e. Commit the configuration.

```
FabricAdmin:switch>cryptocfg --commit
```

Upon commit, redirection zones are created for target port 1, host port 1 and target port 2, host port 2. These redirection zones include the virtual target VT1 for CTC1, the virtual initiator VI1 for host port 1, the virtual target VT2 for CTC2 and the virtual initiator VI2 for host port 2. At this stage, the host loses access to all LUNs until the LUNs are explicitly added to the Crypto Target containers.

4. Discover the LUNs. Perform steps 4 a. through c. to discover the LUNs for ALL CryptoTarget containers in sequence. Refer to the section [“Discovering a LUN”](#) on page 109 for details on the LUN discovery process and a command output example.
 - a. On the encryption switch 1 (the group leader), enter the `cryptocfg --discoverLUN` for the container CTC1. The command output displays the LUNs present in the target as exposed from target port 1 and as seen by host port1, the LUN Number, host port1 WWN, and the LUN Serial Number.

```
FabricAdmin:switch>cryptocfg --discoverLUN CTC1
```
 - b. On the encryption switch 2, enter the `cryptocfg --discoverLUN` for the container CTC2. The command output displays the LUNs present in the target as exposed from target port and as seen by host port 2, the LUN Number, host port1 WWN, and the LUN Serial Number.

```
FabricAdmin:switch>cryptocfg --discoverLUN CTC2
```
 - c. Review the output of the LUN discovery to ensure that the LUN serial number for ALL LUNs are the same as seen from target-port 1 to host-Port 1 path and from target-port 2 to host-port 2. Identical LUN serial numbers validate the multi-path configuration.
5. Configure the LUN for all CryptoTarget containers in sequence by adding the LUN to each CryptoTarget container with identical policy settings. Refer to the sections [“Configuring a Crypto LUN”](#) on page 110 and [“Crypto LUN parameters and policies”](#) on page 112 for more information.

- a. Add the LUN to the CryptoTarget container CTC1 with policies.

```
FabricAdmin:switch>cryptocfg --add -LUN CTC1 0 <Host Port1 WWN> \  
<Host NWWN> -lunstate cleartext -encryption_format native -encrypt \  
-enable_encexistingdata -enable_rekey 10
```

- b. Add the same LUN to the CryptoTarget container CTC2. Use exactly the same LUN state and policy settings that you used for the LUN added to CTC1.

```
FabricAdmin:switch>cryptocfg --add -LUN CTC2 0 <Host Port1 WWN> \  
<Host NWWN> -lunstate cleartext -encryption_format native -encrypt \  
-enable_encexistingdata -enable_rekey 10
```

NOTE

The LUN policies must be exactly the same on both CTC1 and CTC2. Failure to do so results in undefined behavior and data corruption.

6. Validate the LUN policies for all containers. Display the LUN configuration for ALL CryptoTarget containers to confirm that the LUN policy settings are the same for all CryptoTarget containers.

```
FabricAdmin:switch>cryptocfg --show -LUN CTC1 0 <Host Port1 WWN> -cfg
FabricAdmin:switch>cryptocfg --show -LUN CTC2 0 <Host Port2 WWN> -cfg
```

Example:

```
FabricAdmin:switch>cryptocfg --show -LUN CTC10 10:00:00:00:c9:51:39:eb -cfg
EE node:                10:00:00:05:1e:53:4c:8e
EE slot:                0
Target:                50:06:01:68:41:e0:b6:0c 50:06:01:60:c1:e0:b6:0c
VT:                    20:00:00:05:1e:53:4c:9e 20:01:00:05:1e:53:4c:9e
Number of host(s):     1
Configuration status:  committed
Host:                  10:00:00:00:c9:51:39:eb 20:00:00:00:c9:51:39:eb
VI:                    20:02:00:05:1e:53:4c:9e 20:03:00:05:1e:53:4c:9e
LUN number:           0x0
LUN type:              disk
LUN status:            0
Encryption mode:      encrypt
Encryption format:    native
Encrypt existing data: enabled
Rekey:                 enabled
Key ID:                not available
Key life:              47 (minute)
Rekey status:         0
Operation succeeded.
```

7. Commit the LUN configuration.

```
FabricAdmin:switch>cryptocfg -commit
```

NOTE

There is a 25 LUN transaction limit per commit operation. Make sure to issue commit after adding 24 LUNs (12 LUNs to each CTC) so that the LUNs are added to both Crypto Target containers before commit is issued.

3 Tape pool configuration

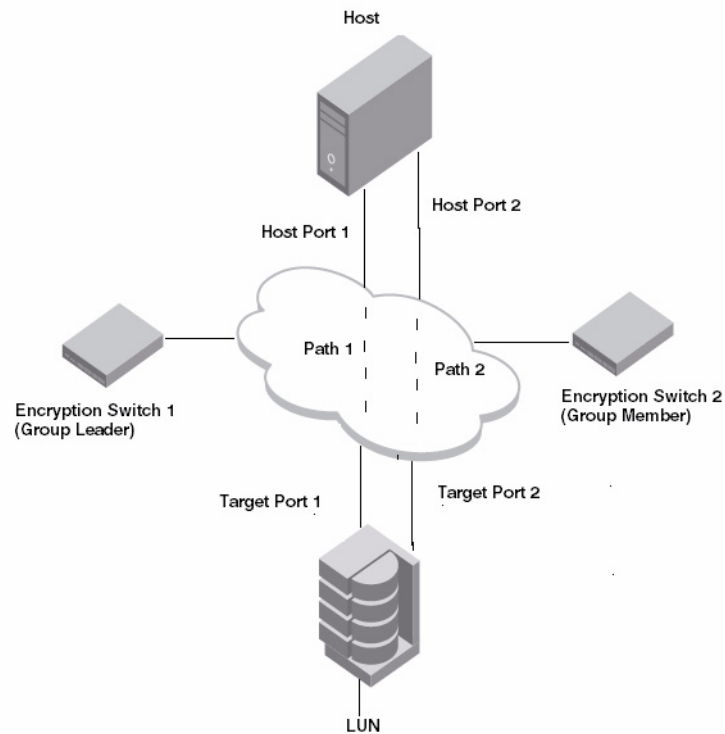


FIGURE 57 A LUN accessible through multiple paths

Tape pool configuration

Tape pools are used by tape backup application programs to group all configured tape volumes into a single backup to facilitate their management within a centralized backup plan. A tape pool is identified by either a name or a number, depending on the backup application. Tape pools have the following properties:

- They are configured and managed per encryption group at the group leader level.
- All encryption engines in the encryption group share the same tape pool policy definitions.
- Tape pool definitions are not needed to read a tape. The tape contains enough information (encryption method and key ID) to enable any encryption engine to read the tape. Tape pool definitions are only used when writing tapes.
- Tape pool names and numbers must be unique within the encryption group.
- If a given tape volume belongs to a tape pool, tape pool-level policies (defaults or configured values) are applied and override any LUN-level policies.
- Tape drive (LUN) policies are used if no tape pools are created or if a given tape volume does not belong to any configured tape pools.

NOTE

Tape pool configurations must be committed to take effect. There is an upper limit of 25 on the number of tape pools you can add or modify in a single commit operation. Attempts to commit a configuration that exceeds this maximum fails with a warning. Note that there is also a five second delay before the commit operation takes effect.

Tape pool labeling

Tape pools may be identified by either a name or a number depending on your backup application. Numbers are always entered and displayed in hex notation. Names and numbers are independent; it is possible to have one tape pool with the name "ABC" and another with the hex number "ABC".

The following rules apply when creating a tape pool label:

- Tape pool names are limited in length to 63 characters. They may contain alphanumeric characters, and in some cases, underscores (_) and dashes (-).
- Tape pool numbers are limited to eight hex digits. Valid characters for tape pool numbers are 0-9, A-F, and a-f.
- The tape pool label created on the encryption switch or blade *must be the same* Tape pool label that is configured on the tape backup application. It is recommended that you check for any labeling restrictions specific to your backup application before creating a tape pool on the encryption switch.

NOTE

Check your backup application before choosing a label. Some applications, such as NetWorker do not support underscore characters in tape pool labels. You may create the tape pool on the application first to determine possible naming restrictions, then use the label generated by the backup application to create the tape pool on the encryption switch or blade.

- A tape pool must first be created on the encryption switch or blade before you can label the tape media and assign them to the tape pool. Failure to observe this sequence invalidates tape pool-level settings and policies, and default LUN-level settings are applied to the tape media.

CommVault Galaxy labeling

CommVault uses a “storage policy” for each backup. When configuring a tape pool to work with CommVault Galaxy, you first create a storage policy on CommVault and then use the *storage_policy_id* (*sp_id*) as the label when creating the tape pool on the encryption switch or blade.

1. Create a storage policy for the backup using the CommVault application. Refer to the product documentation for instructions.
2. Open CommCellExplorer Views by selecting **Start > Programs > Microsoft SQL Server 2005 > SQL ServerManagement Studio**.
3. Expand the tree in the left pane and navigate to the following location:
Comm_serve_computer_name \database_instance_name > Databases > CommServ > Views.
4. Edit the **dbo.CommCellStoragePolicyquery** as follows:

3 Tape pool configuration

- a. Right-click the view and select **Edit**.
- b. Add the following (sp_id= ARG.id) as follows:
 - SELECT Distinct
 - storagepolicy= ARG.name,
 - sp_id= ARG.id,
5. Save the query by selecting **File > Save SQLQuery1.sql**
6. Execute the query by right-clicking the query window and selecting **Execute**.
7. Open the **dbo.CommCellStoragePolicy** view.
8. Right-click the view **dbo.CommCellStoragePolicy** and select **Open View**.
9. Note down the *sp_id* for the storage policy you created. This is the *sp_id* that you use as the tape pool label when creating the tape pool on the encryption switch or blade.

NetBackup labeling

NetBackup uses numbers to label tape pools. If you are using NetBackup as your application, follow these steps to obtain the tape pool number.

1. Log into the NetBackup application Windows host.
2. Select **Start > run**, and type **cmd** in the dialog box.
3. Navigate to C:\Program Files\VERITAS\Volmgr\bin and enter the following command:

```
C:\Program Files\VERITAS\Volmgr\bin>vmpool -listall
=====
pool number: 0
pool name:   None
description: the None pool
pool host:   ANYHOST
pool user:   ANY
pool group:  NONE
=====
```

4. Note down the pool number. This is the number that you use as the tape pool number when creating the tape pool on the encryption switch or blade.

NetWorker labeling

If you use NetWorker as your backup application, be aware of possible naming restrictions. For example, NetWorker does not allow underscore characters in tape pool names. To ensure that you can use the same tape pool name on your encryption platform and on your backup application, create the tape pool on NetWorker first before creating the tape pool on your encryption switch.

Creating a tape pool

Take the following steps to create a tape pool:

1. Log into the group leader as FabricAdmin.
2. Create a tape pool by entering the **cryptocfg --create -tapepool** command. Provide a label or numeric ID for the tape pool and specify the encryption policies. For policies not specified at this time, LUN-level settings apply.
 - Set the tape pool policy to either **encrypt** or **cleartext** (default).
 - Set the encryption format to **DF_compatible** or Brocade **native** (default)

NOTE

To encrypt tapes in DataFort-compatible encryption format (both metadata and encryption algorithm), the DataFort-compatible encryption format needs to be set both at the LUN-level (tape drive) and at the tape pool-level. This ensures that the latest version of DataFort (v2.x/3.x or later) can read and decrypt these tapes.

- Optionally set an expiration date in days for the key (default is no expiration). If the **key_lifespan** parameter is set at the tape pool level to other than none (default), the tape value is used over any LUN-level settings. If the **key_lifespan** parameter is not set at the tape level (default of none), LUN level settings apply.

The following example creates a tape pool named “my_tapepool”.

```
FabricAdmin:switch>cryptocfg --create -tapepool -label my_tapepool
Operation succeeded.
```

3. Commit the transaction.

```
FabricAdmin:switch>cryptocfg --commit
Operation succeeded.
```

4. Display the configuration. Enter the **cryptocfg --show -tapepool** command followed by the tape pool number or label and the **-cfg** parameter.

```
FabricAdmin:switch>cryptocfg --show -tapepool -label my_tapepool -stat
Number of tapepool session(s): 1
Tapepool 1:
Tapepool label:          my_tapepool
Encryption mode:         encrypted
Encryption format:       native
Number of sessions:      0
Tape sessions within the pool:
Operation succeeded.
```

5. Configure the tape pool on your backup application with the same tape pool label you used to create the tape pool on the encryption switch or blade. Refer to the manufacturer’s product documentation for instructions.
6. On your backup application, label the tape media to assign to the tape pool. Refer to the manufacturer’s product documentation for instructions.

Deleting a tape pool

This command does not issue a warning if the tape pool being deleted has tape media or volumes that are currently accessed by the host. Be sure the tape media is not currently in use.

1. Log into the group leader as FabricAdmin.
2. Enter the **cryptocfg --delete -tapepool** command followed by a tape pool label or number. Use **cryptocfg --show -tapepool -all** to display all configured tape pool names and numbers.

```
FabricAdmin:switch>cryptocfg --delete -tapepool -label my_tapepool
Operation succeeded.
```

3. Commit the transaction

```
FabricAdmin:switch>cryptocfg --commit
Operation succeeded.
```

Modifying a tape pool

1. Log into the group leader as FabricAdmin.
2. Enter the **cryptocfg --modify -tapepool** command followed by a tape pool label or number. Then specify a new policy, encryption format, or both. The following example changes the encryption format from Brocade native to DF-compatible.

```
FabricAdmin:switch>cryptocfg --modify -tapepool -label my_tapepool
-encryption_format DF_compatible
Operation succeeded.
```

3. Commit the transaction.

```
FabricAdmin:switch>cryptocfg --commit
Operation succeeded.
```

Impact of tape LUN configuration changes

LUN-level policies apply when no policies are configured at the tape pool level. The following restrictions apply when modifying tape LUN configuration parameters:

- If you change a tape LUN policy from **encrypt** to **cleartext** or from **cleartext** to **encrypt**, or if you change the encryption format from Brocade **native** to **DF-compatible** while data is written to or read from a tape backup device, the policy change is not enforced until the current process completes and the tape is unmounted, rewound, or overwritten. This mechanism prevents the mixing of cleartext data to cipher-text data on the tape.
- Make sure you understand the ramifications of changing the tape LUN encryption policy from **encrypt** to **cleartext** or from **cleartext** to **encrypt**. Refer to [“DF-compatibility support for tape LUNs”](#) on page 199 for information on the impact of policy changes when working in DataFort-compatible encryption format.
- You cannot modify the key lifespan value. If you wish to modify the key lifespan, delete and recreate the LUN with a different key lifespan value. Key lifespan values only apply to native-mode pools. When in DF-compatible mode, every new media receives a unique key, matching DataFort behavior.

Impact of tape pool configuration changes

Tape pool-level policies overrule policy configurations at the LUN level, when no policies are configured at the tape pool level. The following restrictions apply when modifying tape pool-level configuration parameters:

- If you change the tape pool policy from **encrypt** to **cleartext** or from **cleartext** to **encrypt** or if you change the encryption format from Brocade **native** to **DF-compatible** while data is written to or read from a tape backup device, the policy change is not enforced until the current process completes and the tape is unmounted, rewound, or overwritten. This mechanism prevents the mixing of cleartext data to cipher-text data on the tape.
- You cannot modify the tape pool label or the key lifespan value. If you wish to modify these tape pool attributes, delete the tape pool and create a new tape pool with a different label and key lifespan. Key lifespan values only apply to native-mode pools. When in DF-compatible mode, every new media receives a unique key, matching DataFort behavior.

Data re-keying

In a re-keying operation, encrypted data on a LUN is decrypted with the current key, re-encrypted with a new key and written back to the same LUN at the same logical block address (LBA) location. This process effectively re-encrypts the LUN and is referred to as “in-place re-keying.”

It is recommended you limit the practice of re-keying to the following situations:

- Key compromise as a result of a security breach.
- As a general security policy to be implemented as infrequently as every 6 months or once per year.

Re-keying is only applicable to disk array LUNs or fixed block devices. There is no re-keying support for tape media. If there is a need to re-encrypt encrypted tape contents with a new key, the process is equivalent to restoring the data from tape backup. You decrypt the data with the old DEK and subsequently back up the tape contents to tape storage, which will have the effect of encrypting the data with the new DEK.

Resource Allocation

Fabric OS 6.2.0 supports a maximum of 12 outstanding sessions per encryption switch or blade with a maximum of two concurrent sessions per OB1 FPGA. This includes both re-key (auto and manual) and first time encryption sessions. Since the virtual targets and virtual initiators are assigned to an OB1, there is an effective limit of two concurrent re-key/encryption sessions per target container and of two concurrent sessions per physical initiator. If your configuration has two containers that are accessed by the same physical initiator, you cannot have more than two concurrent re-key or encryption sessions.

When scheduled re-key or first time encryption sessions exceed the maximum allowable limit, these sessions will be pending and a "Temporarily out of resources" message is logged. Whenever an active re-key or first time encryption session completes, the next pending session is scheduled.

The system checks once every hour to determine, if there are any re-key or first time encryption sessions pending. If resources are available, the next session in the queue is processed. There may be up to an hour lag before the next session in the queue is processed. It is therefore recommended that you do not schedule more than 12 re-key or first time encryption sessions.

Re-keying modes

Re-keying operations can be performed under the following conditions:

- **Offline re-keying** - The hosts accessing the LUN are offline, or host I/O is halted.
- **Online re-keying** - The hosts accessing the LUN are online, and host I/O is active.

Configuring a LUN for automatic re-keying

Re-keying options are configured at the LUN level either during LUN configuration with the `cryptocfg --add -LUN` command, or at a later time with the `cryptocfg --modify -LUN` command.

For re-keying of a disk array LUN, the Crypto LUN is configured in the following way:

- Set LUN policy as either **cleartext** or **encrypt**.
 - If cleartext is enabled (default), all encryption-related options are disabled and no DEK is associated with the LUN. No encryption is performed on the LUN.
 - If the LUN policy is set to encrypt, encryption is enabled on the LUN and all other options related to encryption are enabled. A DEK is generated and associated with the LUN.
- Set the auto re-keying feature with the `cryptocfg --enable_rekey` command and specify the interval at which the key expires and automatic re-keying should take place (*time period in days*) Enabling automatic re-keying is valid only if the LUN policy is set to **encrypt** and the encryption format is Brocade **native**. Refer to the section [“Crypto LUN parameters and policies”](#) on page 112 for more information.
- When using Brocade native mode in LKM installations, manual rekey is highly recommended. If auto rekey is desired, the key expiry date should be configured only when the LUN is created. Never modify the expiry date after configuring a LUN. If you modify the expiry time after configuring the LUN, the expiration date will not update properly.

NOTE

For a scheduled re-keying session to proceed, all encryption engines in a given HA cluster, DEK cluster, or encryption group must be online, and IO sync links must be configured. Refer to the section [“Management port configuration”](#) on page 87 for more information.

1. Log into the group leader as FabricAdmin.
2. Enable automatic re-keying by setting the `-enable_rekey` parameter followed by a time period (in days). The following example enables the automatic re-keying feature on an existing LUN with a 90-day re-keying interval. The data will automatically be re-encrypted every 90 days.

```
FabricAdmin:switch>cryptocfg --modify -LUN my_disk_tgt 0x0 \  
10:00:00:00:c9:2b:c9:3a -enable_rekey 90  
Operation Succeeded
```

3. Commit the configuration.

```
FabricAdmin:switch>cryptocfg --commit  
Operation Succeeded
```


Initiating a manual re-key session

If auto re-keying is disabled, you can initiate a re-keying session manually at your own convenience. All encryption engines in a given HA cluster, DEK cluster, or encryption group must be online for this operation to succeed. The manual re-keying feature is useful when the key is compromised and you want to re-encrypt existing data on the LUN before taking action on the compromised key.



CAUTION

Do not commit this operation if there are any changes pending for the container in which the re-key was started. If you attempt to do this, the system displays a warning stating that the encryption engine is busy and a forced commit is required for the changes to take effect. A forced commit in this situation will halt any re-key that is in-progress (in any container) and corrupt any LUN that is running re-key at the time. There is no recovery for this type of failure.

1. Log into the group leader as FabricAdmin.
2. Do LUN discovery by issuing the `cryptocfg -discoverLUN` command before issuing the `manual_rekey` command to avoid a potential I/O timeout because of a path state change at the host.
3. Ensure that all encryption engines in the HA cluster, DEK cluster, or encryption group are online by issuing the `cryptocfg --show -groupmember -all` command.
4. Enter the `cryptocfg --manual_rekey` command. Specify the CryptoTarget container name, the LUN number and the initiator PWWN.

```
FabricAdmin:switch>cryptocfg --manual_rekey my_disk_tgt 0x0\
10:00:00:05:1e:53:37:99
Operation Succeeded
Please check the status of the operation using "cryptocfg --show -rekey"
```

5. Check the status of the re-keying session.

```
FabricAdmin:switch> cryptocfg --show -rekey -all
Number of rekey session(s):1

Container name:my_disk_tgt
EE node:10:00:00:05:1e:53:8b:15
EE slot:0
Target:29:af:00:11:0d:03:00:04 29:af:00:11:0d:03:00:04
Target PID:030e04
VT: 20:14:00:05:1e:53:74:fd 20:14:00:05:1e:53:74:fd
VT PID:5e3201
Host: 10:00:00:05:1e:53:37:99 20:00:00:05:1e:53:37:99
Host PID:030a00
VI: 20:20:00:05:1e:53:74:fd 20:21:00:05:1e:53:74:fd
VI PID:5e3301
LUN number:0x0
LUN serial number:600110D0000400000000000040000000004000000000000000
Rekey session number:5
Percentage complete:10
Rekey state:Write Phase
Rekey role:Primary/Active
Block size:512
Number of blocks:909312
Current LBA:818704
Operation succeeded.
```

Suspension and resumption of re-keying operations

A re-key may be suspended or fail to start for several reasons:

- The LUN goes offline or the encryption switch fails and reboots. Re-key operations are resumed automatically when the target comes back online or the switch comes back up. You cannot abort an in-progress re-key operation.
- An unrecoverable error is encountered on the LUN and the in-progress re-key operation halts. The following LUN errors are considered unrecoverable:

```
SenseKey: 0x3 - Medium Error.  
SenseKey: 0x4 - Hardware Error.  
SenseKey: 0x7 - Data Protect.
```
- An unrecoverable error is encountered during the re-key initialization phase. The re-key operation does not begin and a CRITICAL error is logged. All host I/O comes to a halt. All cluster members are notified.
- For any unrecoverable errors that may occur during any other phase of the process, the re-key operation is suspended at that point and a CRITICAL error is logged. All cluster members are notified. Host I/O to all regions of the LUN is halted. Only READ operations are supported for the scratch space region of the LUN used for storing the status block of the re-key operation.

Once all errors have been corrected you have two recovery options:

- Resume the suspended re-key session. All DEK cluster or HA cluster members must be online and reachable for this command to succeed. If successful, this command resumes the re-key sessions from the point where it was interrupted.
 1. Enter the **cryptocfg --resume_rekey** command, followed by the CryptoTarget container name, the LUN number and the initiator PWWN.

```
FabricAdmin:switch>cryptocfg --resume_rekey my_disk_tgt 0x0 \  
10:00:00:05:1e:53:37:99  
Operation Succeeded
```
 2. Check the status of the resumed re-key session.

```
FabricAdmin:switch> cryptocfg --show -rekey -all
```
- Read all data off the LUN and write it to another LUN. In this case, you can cancel the re-key session by removing the LUN from its container and force committing the transaction. Refer to the section [“Removing a LUN from a CryptoTarget container”](#) on page 111 for instructions on how to remove a LUN by force.

First time encryption

First time encryption, also referred to as encryption of existing data, is similar to the re-keying process described in the previous section, except that there is no expired key and the data present in the LUN is cleartext to begin with.

In a first time encryption operation, cleartext data is read from a LUN, encrypted with the current key and written back to the same LUN at the same logical block address (LBA) location. This process effectively encrypts the LUN and is referred to as “in-place encryption.”

Resource allocation

System resources for first time encryption sessions are shared with re-key sessions. There is an upper limit of twelve sessions with two concurrent sessions per target. Refer to the re-key “[Resource allocation](#)” on page 129 section for details.

First time encryption modes

First-time encryption can be performed under the following conditions:

- **Offline encryption** - The hosts accessing the LUN are offline or host I/O is halted while encryption is in process.
- **Online encryption** - The hosts accessing the LUN are online and host I/O is active during the encryption operation.

Configuring a LUN for first time encryption

First time encryption options are configured at the LUN level either during LUN configuration with the `cryptocfg --add -LUN` command, or at a later time with the `cryptocfg --modify -LUN` command.

1. Set the LUN policy to **encrypt** to enable encryption on the LUN. All other options related to encryption are enabled. A DEK is generated and associated with the LUN.
2. Enable first time encryption by setting the `-enable_encexistingdata` parameter. The existing data on the disk is encrypted using the configured DEK.
3. Optionally set the auto re-keying feature with the `cryptocfg --enable_rekey` command and specify the interval at which the key expires and automatic re-keying should take place (*time period in days*). Enabling automatic re-keying is valid only if the LUN policy is set to **encrypt** and the encryption format is Brocade **native**. Refer to the section “[Crypto LUN parameters and policies](#)” on page 112 for more information.

The following example configures a LUN for first time encryption with re-keying scheduled at a 6-month interval. You must commit the operation to take effect.

```
FabricAdmin:switch>cryptocfg --add -LUN my_disk_tgt 0x0 \  
10:00:00:00:c9:2b:c9:3a 20:00:00:00:c9:2b:c9:3a -encrypt \  
-enable_encexistingdata -enable_rekey 180  
Operation Succeeded
```

3 First time encryption

Deployment Scenarios

In this chapter

- Single encryption switch, two paths from host to target. 132
- Single fabric deployment - HA cluster 133
- Single fabric deployment - DEK cluster 134
- Dual fabric deployment - HA and DEK cluster 135
- Multiple paths, one DEK cluster, and two HA clusters 136
- Multiple paths, DEK cluster, no HA cluster 138
- Deployment in Fibre Channel routed fabrics. 139
- Deployment as part of an edge fabric 141
- Deployment with FCIP extension switches. 142
- Data mirroring deployment. 143
- VmWare ESX server deployments 145

Single encryption switch, two paths from host to target

Figure 58 shows a basic configuration with a single encryption switch providing encryption between one host and one storage device over two the following two paths:

- Host port 1 to target port 1, redirected through CTC T1.
- Host port 2 to target port 2, redirected through CTC T2.

Host port 1 is zoned with target port 1, and host port 2 is zoned with target port 2 to enable the redirection zoning needed to redirect traffic to the correct CTC.

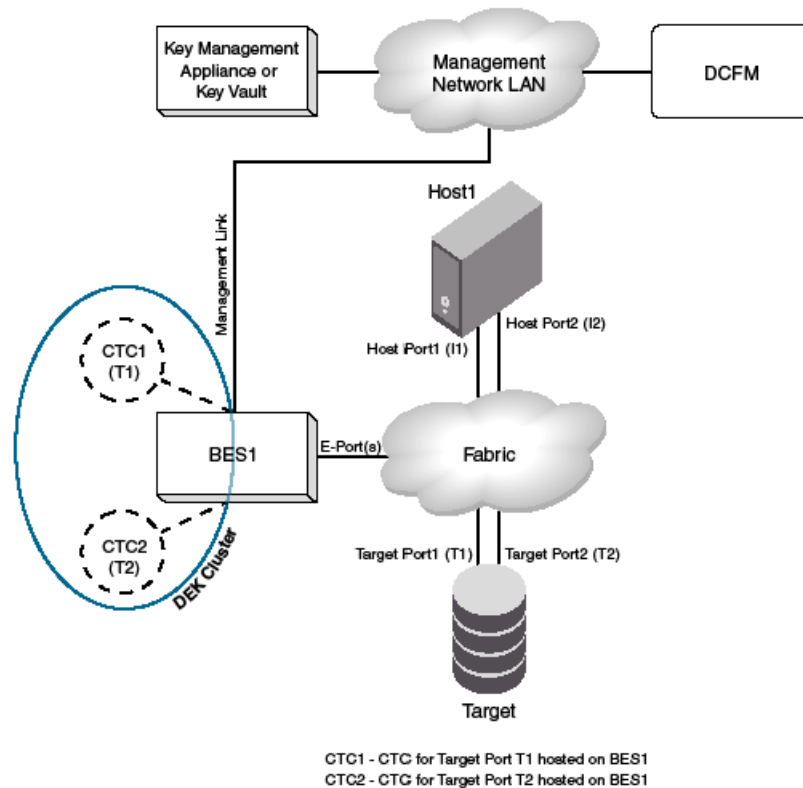


FIGURE 58 Single encryption switch, two paths from host to target

Single fabric deployment - HA cluster

Figure 59 shows an encryption deployment in a single fabric with dual core directors and several host and target edge switches in a highly redundant core-edge topology.

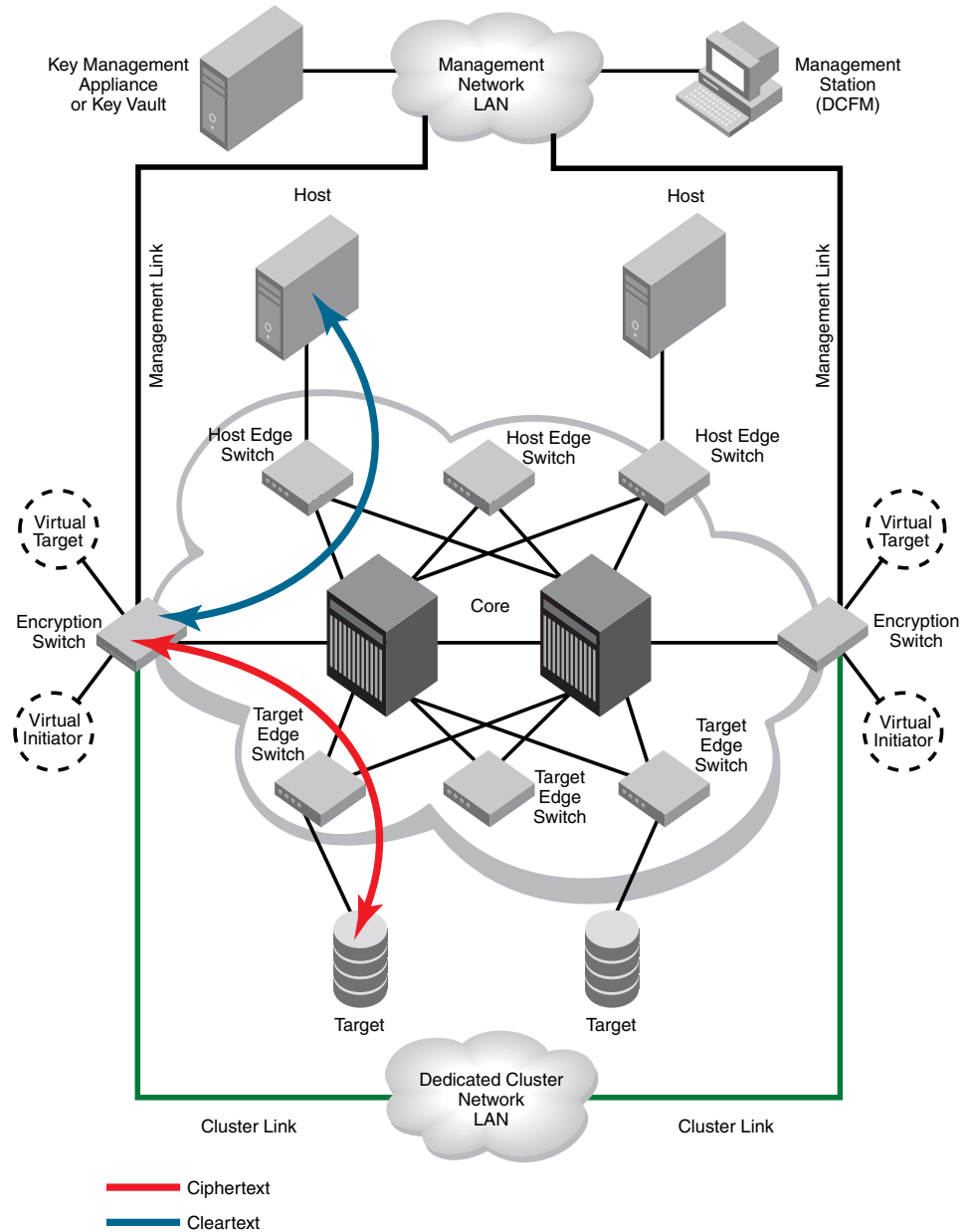


FIGURE 59 Single fabric deployment - HA cluster

4 Single fabric deployment - DEK cluster

In Figure 59, the two encryption switches provide a redundant encryption path to the target devices. The encryption switches are interconnected through a dedicated cluster LAN. The Ge1 and Ge0 gigabit Ethernet ports on each of these switches are attached to this LAN. This LAN connection provides the communication needed to distribute and synchronize configuration information, and enable the two switches to act as a high availability (HA) cluster, providing automatic failover if one of the switches fails, or is taken out of service.

Single fabric deployment - DEK cluster

Figure 60 shows an encryption deployment in a single fabric with two paths between a host and a target device.

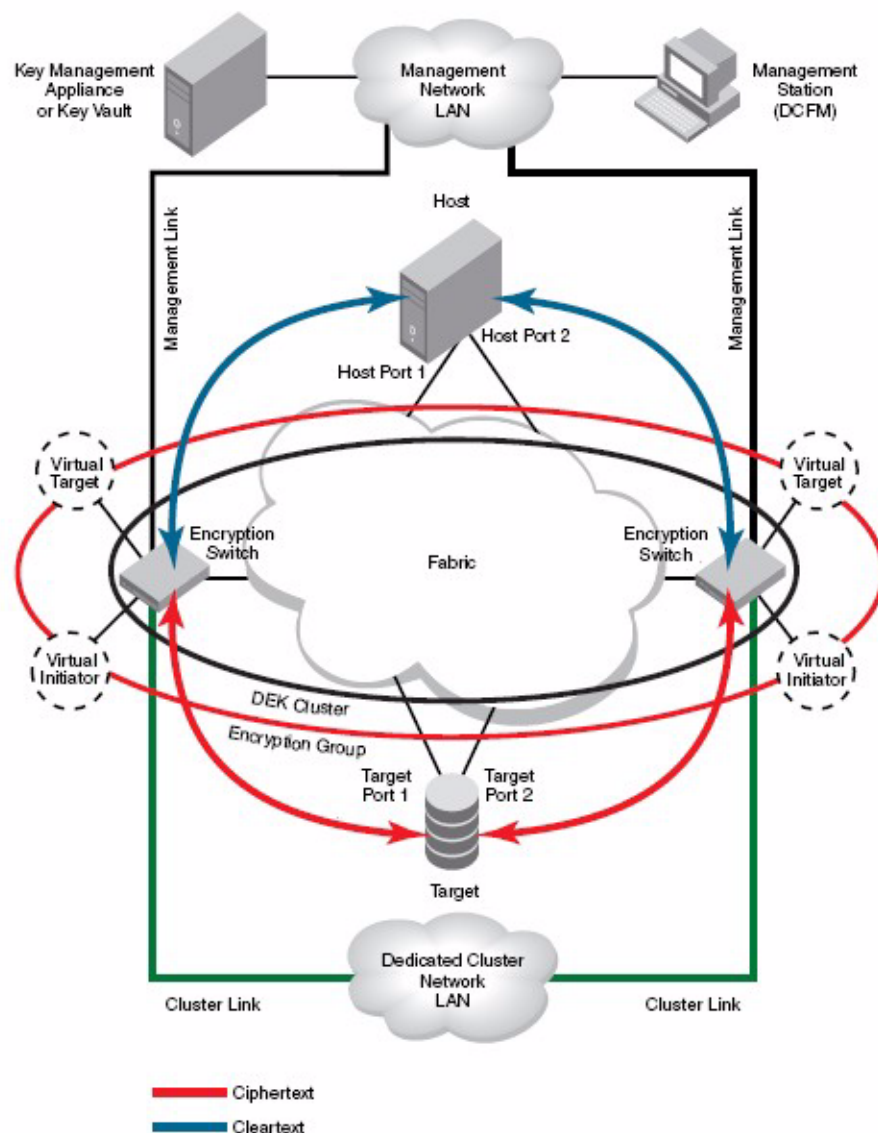


FIGURE 60 Single fabric deployment - DEK cluster

In Figure 60, two encryption switches are required, one for each target path. The path from host port 1 to target port 1 is defined in a CryptoTarget container on one encryption switch, and the path from host port 2 to target port 2 is defined in a CryptoTarget container on the other encryption switch. This forms a DEK cluster between encryption switches for both target paths. Please note that configuring an HA cluster between the two encryption switches in the above configuration is not supported. The DEK cluster handles the target/host path failover along with the failure of either encryption switch.

Dual fabric deployment - HA and DEK cluster

Figure 61 shows an encryption deployment in a dual fabric SAN. Both fabrics have dual core directors and several host and target edge switches in a highly redundant core-edge topology.

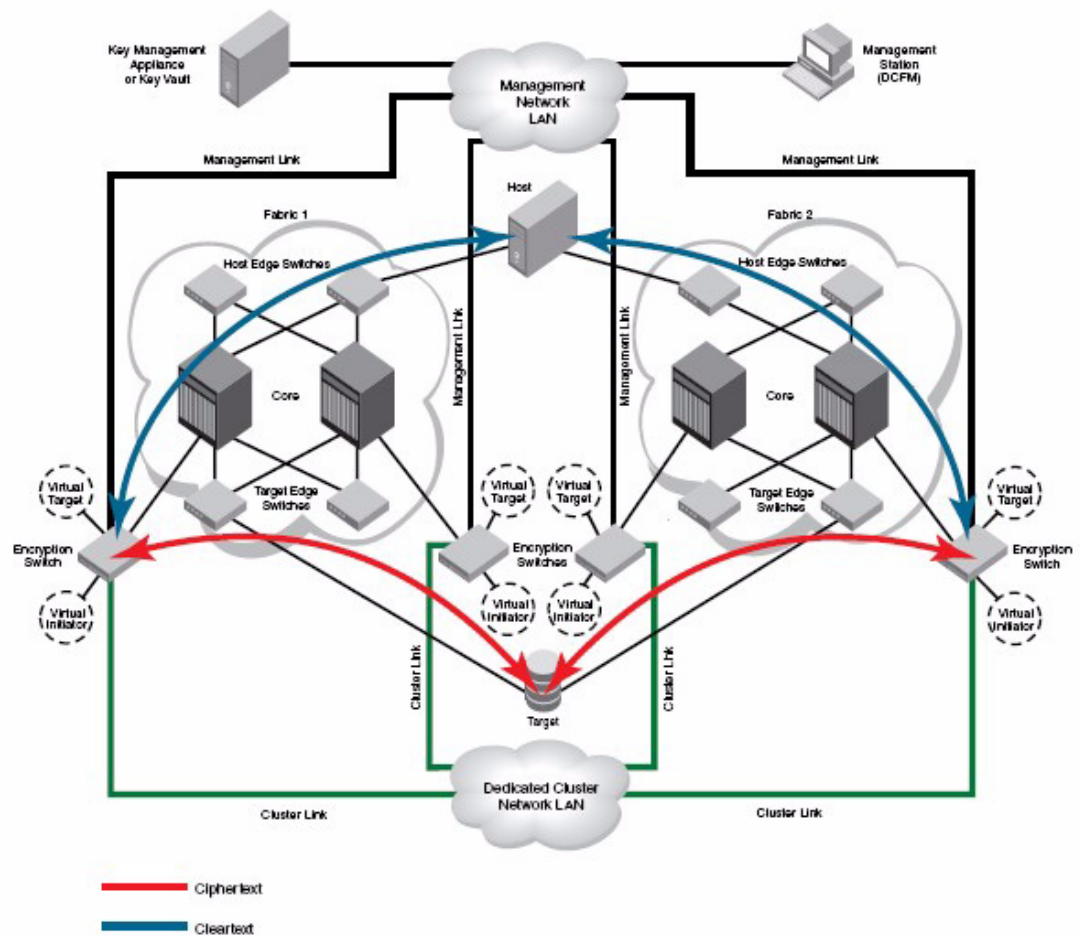


FIGURE 61 Dual fabric deployment - HA and DEK cluster

Figure 61 shows two paths to the target device, one in each fabric. The host also has a path to each fabric. There are two encryption switches in each fabric, interconnected through a dedicated cluster LAN. The Ge1 and Ge0 gigabit Ethernet ports on each of these switches are attached to this LAN. encryption switches 1 and 3 act as a high availability cluster in fabric 1, providing automatic

4 Multiple paths, one DEK cluster, and two HA clusters

failover for the encryption path between the host and target in fabric 1. Encryption switches 2 and 4 act as a high availability cluster in fabric 2, providing automatic failover for the encryption path between the host and target in fabric 2. All four encryption switches provide an encryption path to the same LUN, and use the same DEK for that LUN, forming a DEK cluster.

Multiple paths, one DEK cluster, and two HA clusters

Figure 62 shows a configuration with a DEK cluster that includes two HA clusters, with multiple paths to the same target device.

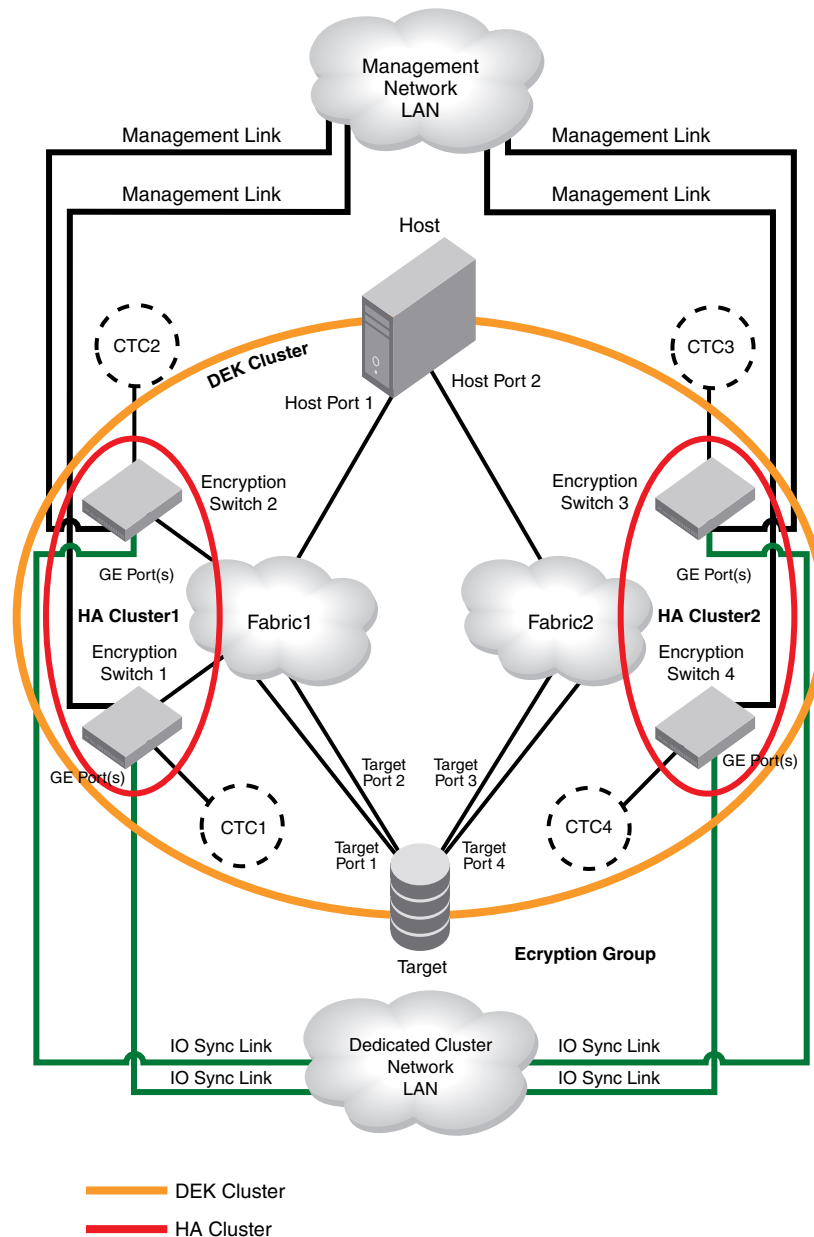


FIGURE 62 Multi-path, DEK cluster and HA cluster

The configuration details shown in [Figure 62](#) are as follows:

- There are two fabrics.
- There are four paths to the target device, two paths in each fabric.
- There are two host ports, one in each fabric.
- Host port 1 is zoned to target port 1 and target port 2 in fabric 1.
- Host port 2 is zoned to target port 3 and target port 4 in fabric 2.
- There are four Brocade encryption switches organized in HA clusters.
- HA cluster 1 is in fabric 1, and HA cluster 2 is in fabric 2.
- There is one DEK cluster, and one encryption group.

Multiple paths, DEK cluster, no HA cluster

Figure 63 shows a configuration with a DEK cluster with multiple paths to the same target device. There is one encryption switch in each fabric.

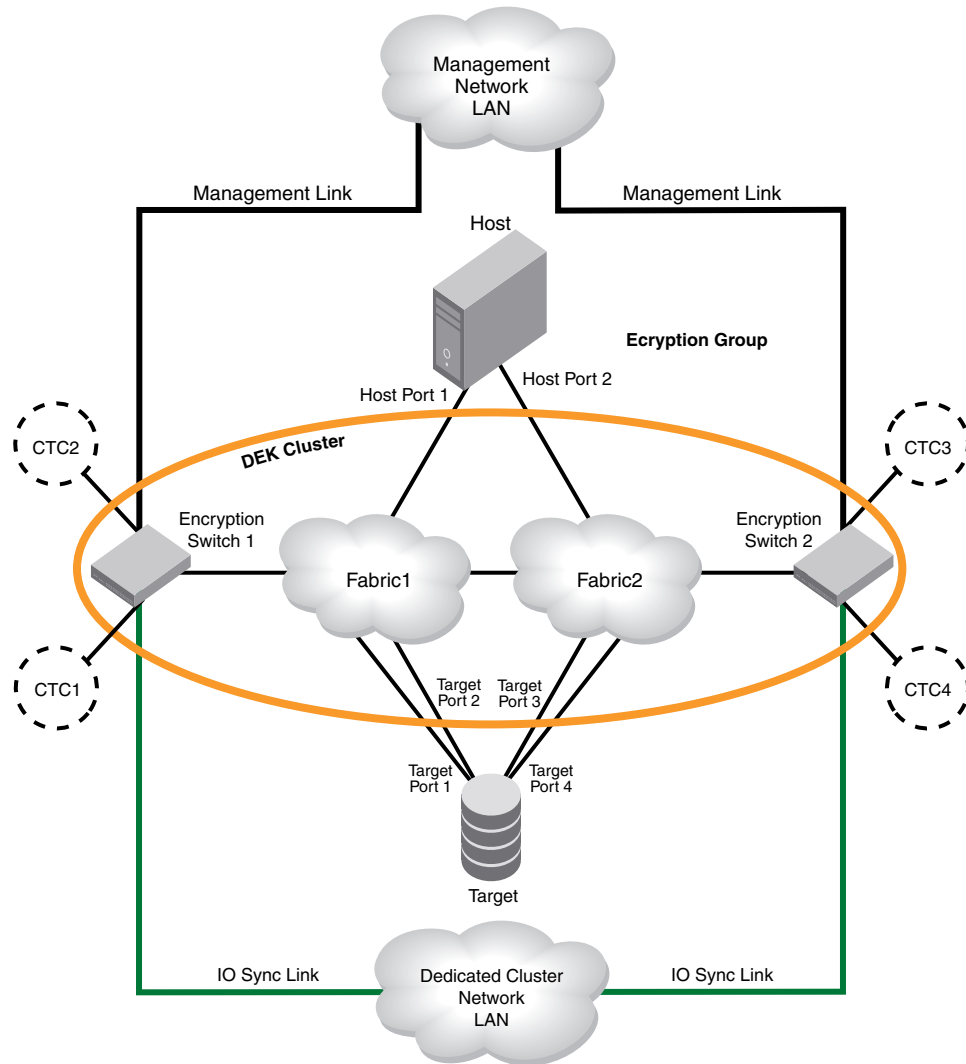


FIGURE 63 Multi-path, DEK cluster, no HA cluster

The configuration details are as follows:

- There are two fabrics.
- There are four paths to the target device, two paths in each fabric.
- There are two host ports, one in each fabric.
- Host port1 is zoned to target port1 and target port2 in fabric 1.
- Host port2 is zoned with target port 3 and target port 4 in fabric 2.
- There are two encryption switches, one in each fabric (no HA cluster).
- There is one DEK Cluster and one encryption group.

Deployment in Fibre Channel routed fabrics

In this deployment, the encryption switch may be connected as part of the backbone fabric to another switch or blade that provides the EX_port connections (Figure 64), or it may form the backbone fabric and directly provide the EX_port connections (Figure 65). The encryption resources can be shared with the host and target edge fabrics using device sharing between backbone and edge fabrics.

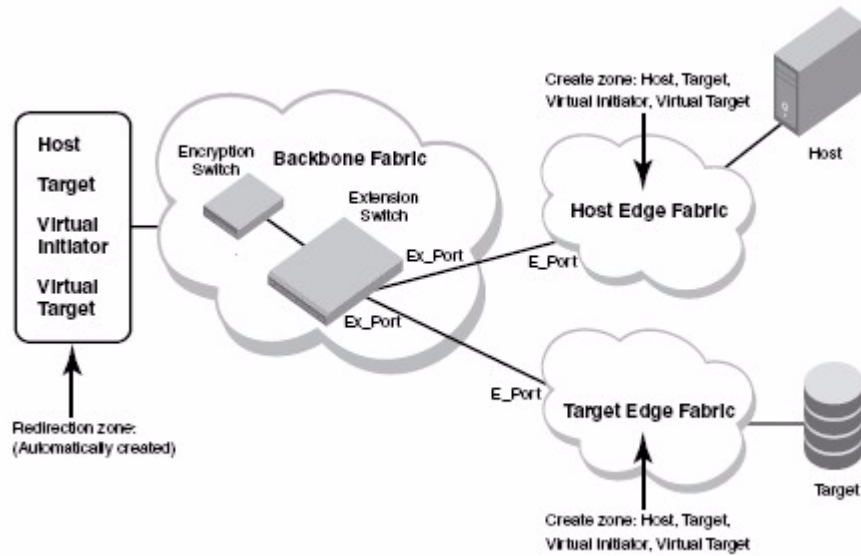


FIGURE 64 Encryption switch connected to FC router as part of backbone fabric

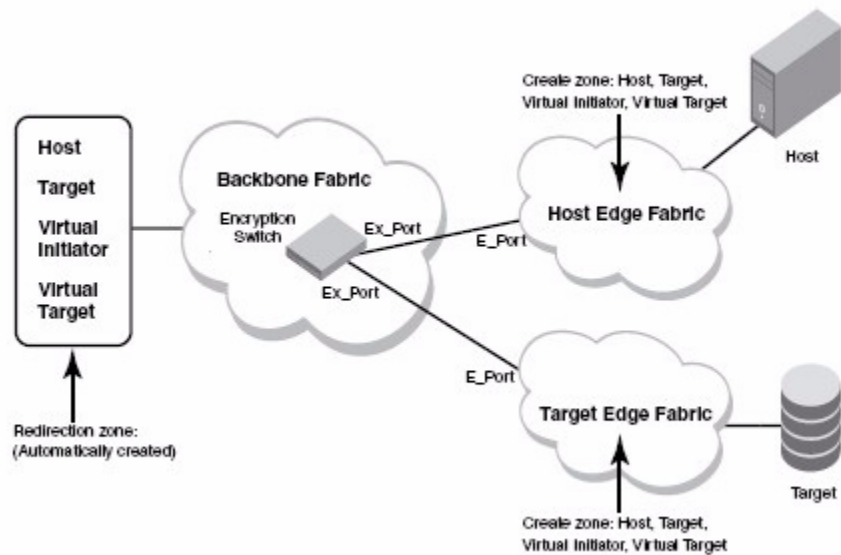


FIGURE 65 Encryption switch as FC router and backbone fabric

4 Deployment in Fibre Channel routed fabrics

The following is a summary of steps for creating and enabling the frame redirection zoning features in the FCR configuration (backbone to edge).

- The encryption device creates the frame redirection zone automatically consisting of host, target, virtual target, and virtual initiator in the backbone fabric when the target and host are configured on the encryption device.
- Create the frame redirection zone consisting of host, target, virtual target, and virtual initiator in both the host and target edge fabrics. The CLI command is **zone --rdcreate [host wwn] [target wwn] [VI wwn] [VT wwn][nonrestartable] [FCR]**. Always specify **nonrestartable** as a policy for creating redirection zones. The VI and VT port WWNs can be obtained by running the **cryptocfg -show -container <crypto container name> -cfg** command on the encryption switch or blade. After the redirection zones are created, commit the configuration with the **cfgsave** command.
- Create the LSAN zone consisting of host, target, virtual target, and virtual initiator in both the backbone fabric and the target edge fabrics. Refer to the *Fabric OS Administrator's Guide* for information about LSANs, LSAN zoning, and Fibre Channel routing (FCR) configurations.

Deployment as part of an edge fabric

In this deployment, the encryption switch is connected to either the host or target edge fabric. The backbone fabric may contain a 7500 extension switch or FR4-18i blade in a 48000 director, DCX, or DCX-4S, or an FCR-capable switch or blade. The encryption resources of the encryption switch can be shared with the other edge fabrics using FCR in the backbone fabric (Figure 66).

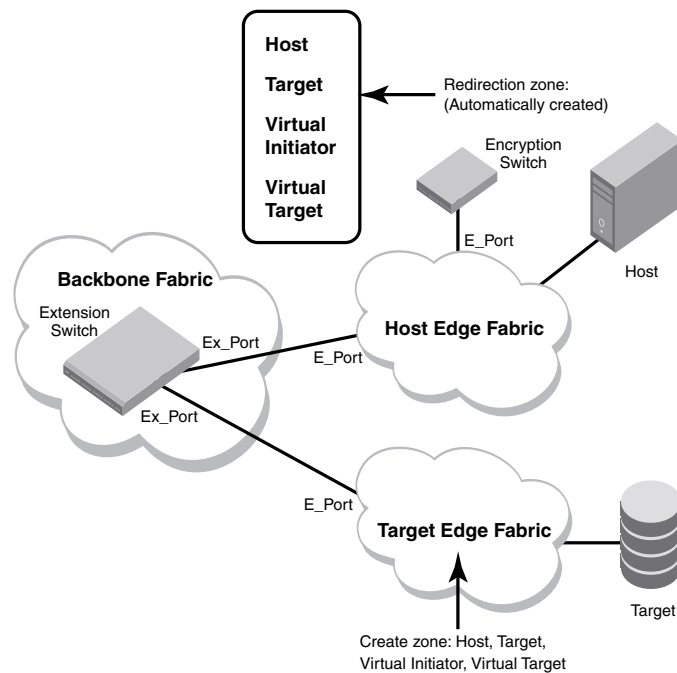


FIGURE 66 Encryption switch as part of an edge fabric

The following is a summary of steps for creating and enabling the frame redirection features in the FCR configuration (edge to edge):

- The encryption device creates the frame redirection zone automatically, consisting of host, target, virtual target, and virtual initiator. when the target and host are configured on the encryption device. In Figure 66, the encryption device is connected to the host edge fabric.
- Create the frame redirection one consisting of host, target, virtual target, and virtual initiator in the target edge fabric. The CLI command is `zone --rdcreate [host wwn] [target wwn] [VI wwn] [VT wwn][nonrestartable] [noFCR]`. Always specify `nonrestartable` as policy for creating redirection zones in case of the encryption device. The VI and VT port WWNs can be obtained by running the `cryptocfg --show -container <crypto container name> -cfg` command on the encryption switch or blade. After the redirection zones are created, commit the configuration with the `cfgsave` command.
- Create the LSAN zone consisting of host, target, virtual target, and virtual initiator in both the backbone fabric and the target edge fabrics. Refer to the *Fabric OS Administrator's Guide* for information about LSANs, LSAN zoning, and Fibre Channel routing (FCR) configurations.

Deployment with FCIP extension switches

Encryption switches may be deployed in configurations that use extension switches or extension blades within a DCX, DCX-4S or 48000 chassis to enable long distance connections. [Figure 67](#) shows an encryption switch deployment in a Fibre Channel over IP (FCIP) configuration. Refer to the *Fabric OS Administrator's Guide* for information about creating FCIP configurations.

When an encryption switch is deployed with an extension switch or blade in the same chassis or fabric, the encryption switch can use the FCIP functionality provided by the extension switch.

In [Figure 67](#), the host is using the remote target for remote data mirroring or backup across the FCIP link. If the encryption services are enabled for the host and the remote target, the encryption switch can take clear text from the host and send cipher text over the FCIP link. For FCIP on the extension switch, this traffic is same as rest of the FCIP traffic between any two FCIP end points. The traffic is encrypted traffic. FCIP provides a data compression option. Data compression should not be enabled on the FCIP link. If compression is enabled on FCIP link, then encrypted traffic going through FCIP compression may not provide the best compression ratio.

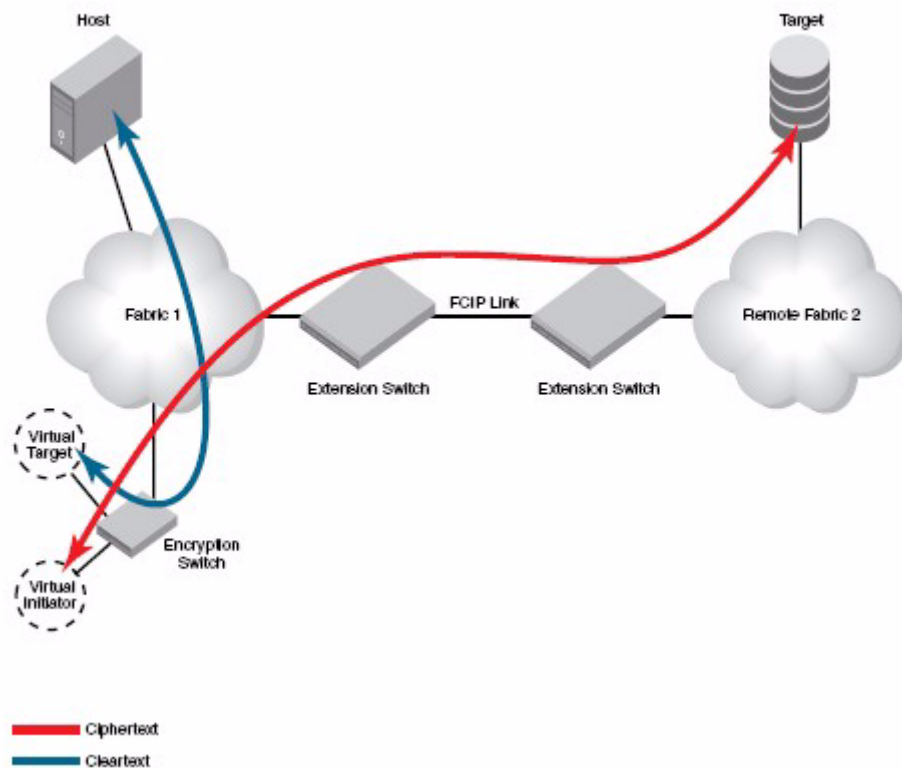


FIGURE 67 FCIP deployment

Data mirroring deployment

Figure 68 shows a data mirroring deployment. In this configuration, the host only knows about target1 and LUN1, and the I/O path to target1 and LUN1. When data is sent to target1, it is written to LUN1, and also sent on to LUN2 for replication. Target1 acts as an initiator to enable the replication I/O path. When an encryption switch is added to the configuration, it introduces another virtual target and LUN, and a virtual initiator in the I/O path in front of target1. The virtual target and LUN provided by the encryption switch is mapped to target1 and LUN1. Data is encrypted and the cipher text is sent to target1, written to LUN1, and replicated on LUN2.

Only one DEK is used to create the cipher text written to both LUNs. A key ID is stored in metadata written to both LUNs. If possible, the metadata is written to every block with the LBA range of 1 to 16. This ensures that the encryption engine will be able to retrieve the correct DEK from the key vault when retrieving data from either LUN1 or LUN2.

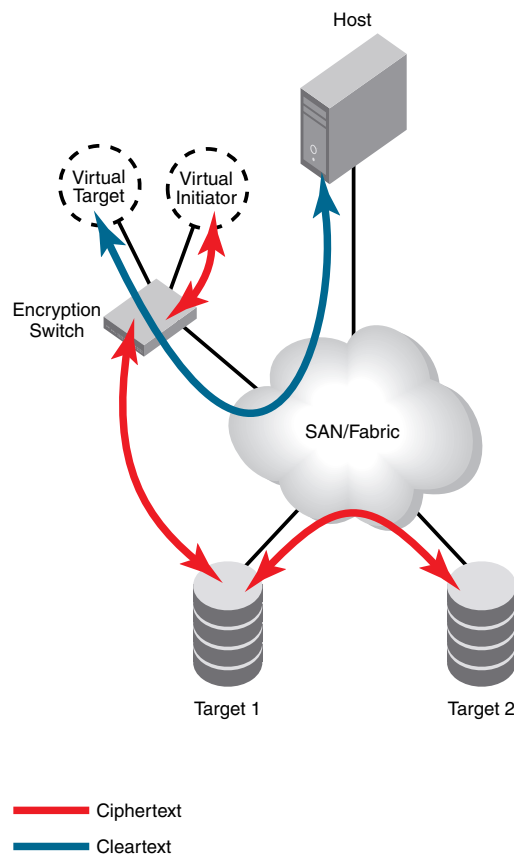


FIGURE 68 Data mirroring deployment

If metadata is not present on the LUN

In very rare cases, metadata may not be present on the LUN. The record archived in the key vault refers only to the primary LUN, and not to the LUN replication. With no metadata present in the replicated blocks, there is no key ID to use to retrieve the DEK from the key vault. User intervention is needed to query the key vault to get the key ID.

1. Map the primary LUN to the replicated or snapshot LUN.
2. Based on the primary LUN information (mainly target WWN, LUN number, or LUN SN), you can query key records from the key vaults. For this, you need to refer to key management system's documentation to find out how to query key records.
3. Identify the key used during the replication or snapshot of the LUN based on the creation and expiry time of the key at the time the LUN was replicated.
4. When the record is identified, provide the Key ID for the key record as input to the LUN addition for this LUN on the encryption switch or blade. This is done from the key management system's user interface. Refer to the user documentation for the key management system.

VmWare ESX server deployments

VM ESX servers may host multiple guest operating systems. A guest operating system may have its own physical HBA port connection, or it may use a virtual port and share a physical HBA port with other guest operating systems.

Figure 69 shows a VmWare ESX server with two guest operating systems where each guest accesses a fabric over separate host ports.

There are two paths to a target storage device:

- Host port 1 to target port 1, redirected through CTC T1.
- Host port 2 to target port 2, redirected through CTC T2.

Host port 1 is zoned with target port 1, and host port 2 is zoned with target port 2 to enable the redirection zoning needed to redirect traffic to the correct CTC.

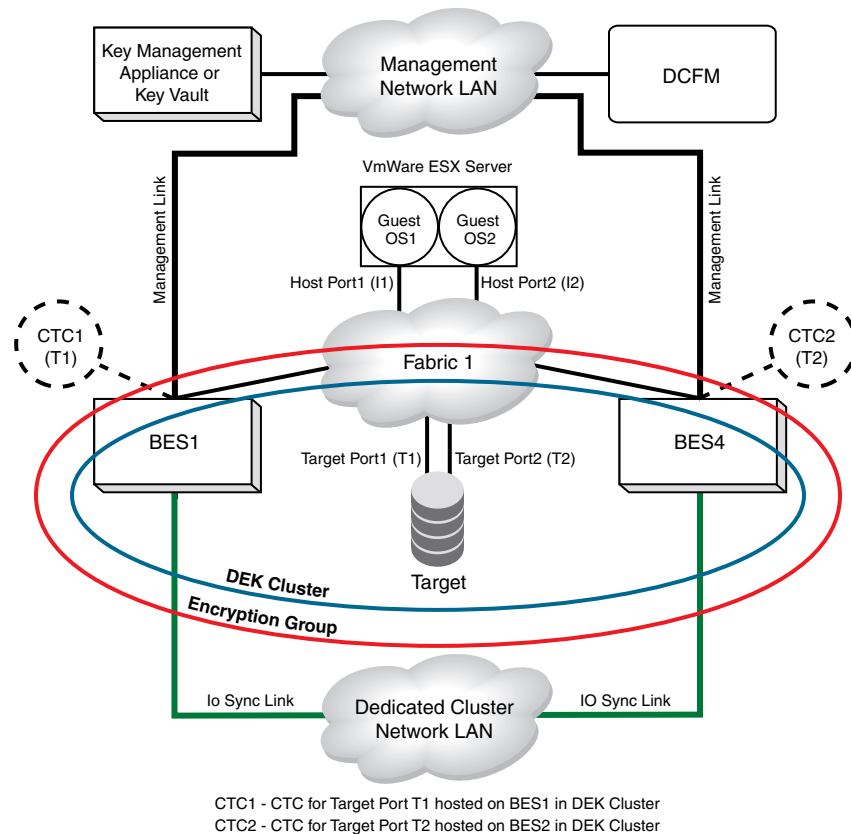


FIGURE 69 VmWare ESX server, One HBA per guest OS

Figure shows a VmWare ESX server with two guest operating systems where two guests access a fabric over a shared port. To enable this, both guests are assigned a virtual port.

There are two paths to a target storage device:

- Virtual host port 1, through the shared host port, to target port 1, redirected through CTC T1.
- Virtual host port 2, through the shared host port, to target port 2, redirected through CTC T2.

In this case, the virtual host port 1 is zoned with target port 1, and the virtual host port 2 is zoned with target port 2 to enable the redirection zoning needed to redirect traffic to the correct CTC.

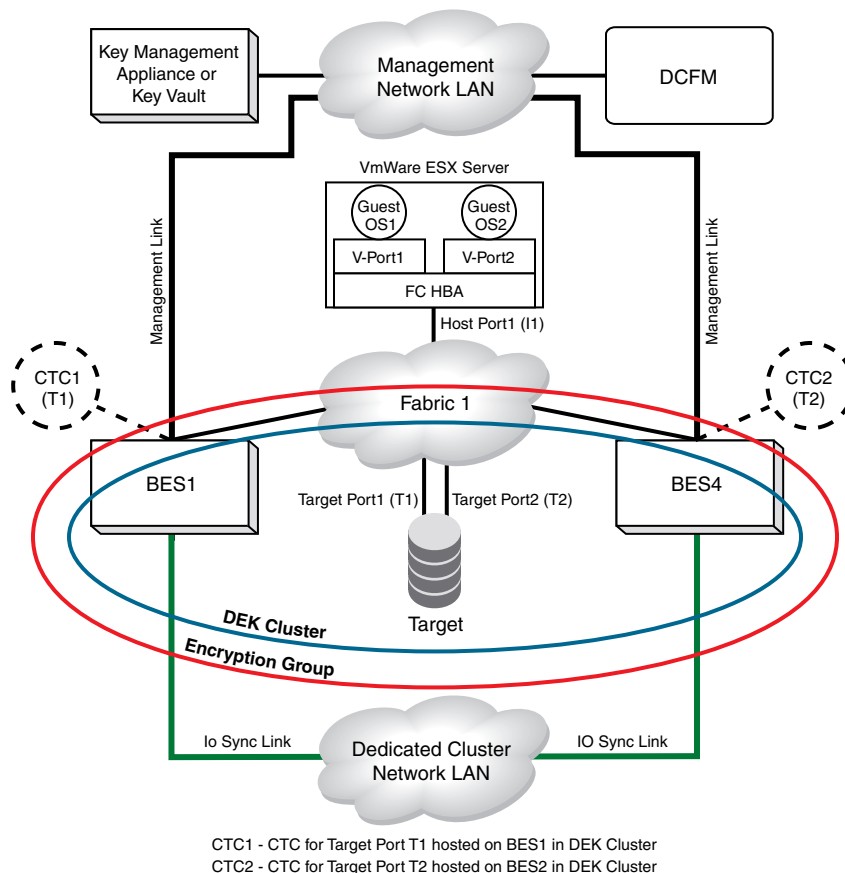


FIGURE 70 VmWare ESX server, One HBA shared by two guest OS

Best Practices and Special Topics

In this chapter

- Firmware download considerations 148
- HP-UX considerations 153
- Enable of a disabled LUN 153
- Disk metadata 153
- Tape metadata 153
- Tape data compression 154
- Tape pools 154
- DF compatibility for tapes 155
- DF compatibility for disk LUNs 155
- Configuring CryptoTarget containers and LUNs 156
- Redirection zones 157
- Deployment with Admin Domains (AD) 157
- Master key usage in RKM and SKM environments 157
- Do not use DHCP for IP interfaces 157
- Ensure uniform licensing in HA clusters 157
- Tape library media changer considerations 158
- Turn off host-based encryption 158
- Avoid double encryption 158
- PID failover 158
- Turn off compression on extension switches 158
- Re-keying best practices and policies 159
- Changing IP addresses in encryption groups 160
- Disabling the encryption engine 160
- Recommendations for Initiator Fan-Ins 161
- Best practices for host clusters in an encryption environment 162
- HA Cluster Deployment Considerations and Best Practices 162

Firmware download considerations

The encryption engine and the control processor or blade processor are reset after a firmware upgrade. Disruption of encryption I/O can be avoided if an HA cluster is configured. If encryption engines are configured in an HA cluster, perform firmware upgrades one encryption engine at a time so that the partner switch in the HA cluster can take over I/O by failover during firmware upgrade. When switches form a DEK cluster, firmware upgrades should also be performed one at a time for all switches in the DEK cluster to ensure that a host MPIO failover path is always available.

Firmware Upgrades and Downgrades

You cannot downgrade to a Fabric OS version release prior to Fabric OS version 6.2.0.

A downgrade to Fabric OS version 6.2.0 results in the loss of the following functionality.

- If an NCKA key vault is being used, a downgrade to v6.2.0 results in a loss of encryption services.
- Fabric OS version 6.2.0 supports only one HP SKM key vault. Registering of a second HP SKM key vault will be blocked.
- Fabric OS version 6.2.0 uses `brcduser1` as a standard user name when creating a Brocade group on SKM. If you downgrade from version 6.3.0 or later to version 6.2.0, the user name is overwritten to `brcduser1`, and the Brocade group user name must be changed to `brcduser1`.
- General guidelines for a firmware upgrade of encryption switches and a DCX or DCX-4S with encryption blades in encryption groups, HA clusters, and DEK clusters are as follows:
 - Upgrade one node at a time.
 - Do not do a firmware upgrade when re-key operations and first time encryption operations are underway.
 - Do not start any manual re-key operations and first time encryption operations during the firmware upgrade process for all nodes in the HA/DEK cluster.
- Guidelines for firmware upgrade of encryption switches and a DCX or DCX-4S with encryption blades deployed in a DEK cluster with two HA clusters:
 - Upgrade nodes in one HA cluster at a time.
 - Within an HA cluster, upgrade one node at a time.
- Guidelines for firmware upgrade of encryption switches and a DCX or DCX-4S with encryption blades deployed in DEK cluster with No HA cluster (each node hosting one path).
 - Upgrade one node at a time.
 - In the case of active/passive arrays, upgrade the node which is hosting the passive path first. Upgrade the node which is hosting active path next. The Host MPIO ensures that I/O fails over and fails back from active to passive and back to active during this firmware upgrade process.
 - In the case of active/active arrays, upgrade order of nodes does not matter, but you still must upgrade one node at a time. The Host MPIO ensures that I/O fails over and fails back from one active path to another active path during this firmware upgrade process.
- All nodes in an encryption group must be at the same firmware level before starting a re-key or first time encryption operation.

Specific guidelines and procedures

The following are specific guidelines for a firmware upgrade of the encryption switch or blade when deployed in HA cluster. The guidelines are based on the following scenario:

- There are 2 nodes (BES1 and BES2) in the HA cluster.
 - Each node hosts certain number of CryptoTarget containers and associated LUNs.
 - node 1 (BES1) needs to be upgraded first.
1. Change the failback mode to manual if it was set to auto by issuing the following command:
cryptocfg –set -failback manual
 2. On node 1 (BES1), disable the encryption engine to force the failover of CryptoTarget containers and associated LUNs onto the HA cluster peer member node 2 (BES2) by issuing the following command.
cryptocfg –disableEE
 3. Make sure that these Crypto Target Containers and LUNs actually failover to node 2 (BES2) in the HA cluster. Check for all LUNs in encryption enabled state on node 2 (BES2). This ensures that I/O also fails over to node 2 (BES2) and continues during this process.
 4. On node 1 (BES1) enable the Encryption Engine, by issuing the following command.
cryptocfg –enableEE
 5. Start firmware download (upgrade) on the node 1 (BES1). Refer to the *Fabric OS Administrator's Guide* if necessary to review firmware download procedures.
 6. After firmware download is complete and node 1 (BES1) is back up, make sure the encryption engine is online.
 7. On node 1 (BES1) initiate manual failback of CryptoTarget containers and associated LUNs from node 2 (BES2) to node 1 (BES1) by issuing the following command.
cryptocfg –failback -EE
 8. Check that Crypto Target Containers and associated LUNs fail back successfully on node 1 (BES1) and host I/O also moves from node 2 (BES2) to node 1 (BES1) and continues during the failback process.
 9. To upgrade node 2 (BES2), Repeat steps 2 to 8.
 10. After all nodes in the Encryption Group have been upgraded, change back the failback mode to auto from manual, if required by issuing the following command.
cryptocfg –set -failback auto

Configuration upload and download considerations

Important information is not included when you upload a configuration from an encryption switch or blade. Extra steps are necessary before and after download to re-establish that information. The following sections describe what information is included in a upload from an encryption group leader and encryption group member load, what information is not included, and the steps to take to re-establish the information.

Configuration Upload at an encryption group leader node

A configuration upload performed at an encryption group leader node contains the following:

- The local switch configuration.
- Encryption group-related configuration.
- The encryption group-wide configuration of Crypto Targets, disk and tape LUNs, tape pools, HA clusters, security, and key vaults.

Configuration upload at an encryption group member node

A configuration upload at an individual encryption group member node contains the following

- The local switch configuration.
- Encryption group-related configuration.

Information not included in an upload

The following certificates will be not be present when the configuration is downloaded.

- External certificates imported on the switch:
 - key vault certificate
 - peer node/switch certificate
 - authentication card certificate
- Certificates generated internally:
 - KAC certificate
 - CP certificate
 - FIPS officer and user certificates

The Authentication Quorum size is included in the configuration upload for read-only purposes, but is not set by a configuration download.

Steps before configuration download

The configuration download does not have any certificates, public or private keys, master key, or link keys included. Perform following steps prior to configuration download to generate and obtain the necessary certificates and keys:

1. Use the following commands to initialize the encryption engine

```
cryptocfg -InitNode
cryptocfg -initEE
cryptocfg -regEE
```

Initializing the switch generates the following internal certificates:

- KAC certificate
 - CP certificate
 - FIPS officer and user certificates
2. Import peer nodes/switches certificates onto the switch prior to configuration download.
 3. Import key vault certificates onto switch prior to configuration download. Refer to [Appendix D, "Supported Key Management Systems"](#) for instructions specific to the key vault you are using.
 4. Create an encryption group with same name as in configuration upload information for the encryption group leader node.
 5. Import Authentication Card Certificates onto the switch prior to configuration download.

Configuration download at the encryption group leader

The configuration download contains the encryption group-wide configuration information about Crypto Targets, disk and tape LUNs, tape pools, HA clusters, security, and key vaults. The encryption group leader first applies the encryption group-wide configuration information to the local configuration database and then distributes the configuration to all members in the encryption group. Also any layer-2 and switch specific configuration information is applied locally to the encryption group leader.

Configuration download at an encryption group member

Switch specific configuration information pertaining to the member switch or blade is applied. Information specific to the encryption group leader is filtered out.

Steps after configuration download

For all key vaults except LKM, restore or generate and backup the master key. In cluster environments, the master key is propagated from group leader node.

1. Use the following command to enable the encryption engine.

```
cryptocfg --enableEE [slot num]
```

2. Commit the configuration.

```
cryptocfg --commit
```

3. If there are containers that belonged to the old encryption switch or blade, then after **configdownload** is run, use the following command to change the ownership of containers to the new encryption switch or blade, assuming the host and target physical zone exists.

```
cryptocfg -replace <old EE WWN> <new EE WWN>
```

4. Commit the configuration.

```
cryptocfg --commit
```

5. Use the following command to check if the switch or blade has the master key.

```
cryptocfg --show -groupmember <switch WWN>
```

6. If a master key is not present, restore the master key from backed up copy. Procedures will differ depending on the backup media used (from recovery smart cards, from the key vault, from a file on the network or a file on a USB-attached device). If new master key needs to be generated, generate the master key and back it up.

For LKM key vaults, establish the trusted link with the LKM appliance. Refer to [Appendix D, "Supported Key Management Systems"](#) for instructions.

If authentication cards are used, set the authentication quorum size from the encryption group leader node, after importing and registering the necessary number of Authentication Card certificates.

HP-UX considerations

The HP-UX OS requires LUN 0 to be present. LUNs are scanned differently based on the type value returned for LUN 0 by the target device.

- If the type is 0, then HP-UX only scans LUNs from 0 to 7. That is the maximum limit allowed by HP-UX for device type for type 0.
- If the type is 0xC, then HP-UX scans all LUNs.

Best practices are as follows:

- Create a cryptoTarget container for the target WWN.
- Add the HP-UX initiator WWN to the container.
- Issue the discover LUN CLI command on the container to discover the LUNs present in the target.
- Based on the LUN list returned as part of LUN discovery, add the LUN 0 if LUN 0 is present in the target (which is usually the case).

Enable of a disabled LUN

When Metadata is found on the LUN, but current LUN state is indicated as cleartext or is being converted from encrypt to cleartext, the LUN is disabled and the LUN status displayed by the LUN Show CLI command is **Encryption Disabled <Reason Code>**.

The disabled LUN can be enabled by the enable LUN command.

```
cryptocfg --enable -LUN <crypto target container name> <LUN Num> <InitiatorPWWN>
```

Disk metadata

If possible, thirty-two bytes of metadata are added to every block in LBA range 1 to 16 for both the native Brocade format and DF-compatible formats. This metadata is not visible to the host. The Host I/Os for the metadata region of the LUN are handled in the encryption switch software, and some additional latency should be expected.

Tape metadata

One kilobyte of metadata is added per tape block for both the native Brocade format and DF-compatible formats. Tape block size (as configured by host) is modified by the encryption device to accommodate 1K metadata per block. A given tape can have a mix of compressed and uncompressed blocks. Block lengths are as follows.

Encrypted/Compressed Tape Block Format	Compressed and encrypted tape block data + 1K metadata + ASCII 0 pad = block length of tape.
Encrypted Tape Block Format (No Compression)	Encrypted tape block data + 1K metadata = block length of tape.

Tape data compression

Data is compressed by the encryption switch or blade before encrypting only if the tape device supports compression, and compression is explicitly enabled by the host backup application. That means if the tape device supports compression, but is not enabled by the host backup application, then compression is not performed by the encryption switch or blade before encrypting the data. However, if the backup application turns on compression at the tape device and does not turn it off before logout or after the backup or restore operation is complete, and a second host backup application starts using the same tape device and does not explicitly turn off compression, compression will still be on when the encryption switch or blade issues a Mode Sense command to find target device capabilities, and compression is used. In other words, if the host backup application does not turn off compression on the target, the encryption switch or blade uses the compression feature of the target. Conversely, if the tape device does not support compression, the encryption switch or blade does not perform compression before encrypting the data. The same rules apply for decompression.

Data is compressed, encrypted and padded with ASCII 0 to the tape block length to simplify handling at the encryption device. It is assumed that a tape target with compression enabled will be unable to compress the seemingly random encrypted data, but will greatly compress the padded zero data that follows. Compressing data at the encryption device in conditions other than above does not create any additional space savings on the tape media.

Tape pools

When a new tape pool needs to be created, the following steps must be performed:

- Configure the tape pool with a maximum of 64 bytes of tape pool label first on the encryption device. The tape pool label configured on the encryption device must be an exact match to the tape pool label configured on the tape backup application.
- Set the policies (such as encrypt or cleartext), format (such as native Brocade format or DF-compatible), and optionally specify a key life span for the tape pool.

Tape pools are unique across an encryption group. Tape pool configuration takes precedence over LUN level configuration.

Tape pool configuration is used only when labeling of tape media is done on the first write for the tape media. After tape labeling is done and metadata written, the tape pool configuration is no longer used. Tape pool configuration is not required for restoring data from the encrypted tape belonging to the tape pool, because the key ID is present in the metadata.

When the tape pool label configured on the encryption device does not match with any label that the backup application sends as part of the first write (tape labeling) to the tape media, the tape pool level policies are ignored and default LUN level policies are applied.

Tape block zero handling

The block zero of the tape media is not encrypted and the data in the block zero is sent as cleartext along with the block zero metadata header prefixed to the data to the tape device.

Tape key expiry

When the tape key expires in the middle of a write operation on the tape, the key is used for the duration of any write operation to append the data on the tape media. On any given tape medium, the same key is used for all written blocks, regardless of the time in between append operations.

With the exception of native pools, whenever you rewind a tape and write to block zero, a new key will be generated, unique to that tape. Only with native pools will the same key be used to write to multiple media. This key has a user-determined lifespan, which applies to the elapsed time between write operations to new tapes (after rewind). Key expiration does not apply to append operations, no matter how long in the future.

Key expiration never applies to reads.

DF compatibility for tapes

Only DF version 2.x- and 3.x-compatible NetApp DataFort (DF) tape metaheaders and block formats are supported for reading, decrypting, and decompressing the tapes.

Only DF version 2.x- and 3.x-compatible tape block formats and metaheaders are supported for writing and encrypting tapes in DF-compatible format.

A DF-compatible license is required.

DF compatibility for disk LUNs

Most versions of NetApp DataFort (DF) disk metaheaders and block formats are supported for reading, decrypting, and decompressing the disk LUNs. DF 1.x version disks are not supported for reading.

Only DF version 3.x-compatible disk block formats and metaheaders are supported for writing and encrypting disk LUNs in DF-compatible format. A DF-compatible license is required.

Configuring CryptoTarget containers and LUNs

The following are best practices to follow when configuring CryptoTarget containers and crypto LUNs:

- Host a target port on only one encryption switch, or one HA cluster. All LUNs visible through the target port are hosted on the same encryption switch, and are available for storing cipher text.
 - Be sure all nodes in a given DEK or HA cluster are up and enabled before creating an encrypted LUN. If a node in the DEK or HA cluster is down, or the encryption engine is down or not enabled when an encrypted LUN is added to the CryptoTarget container, write operations will hang when writing metadata to the LUN, and I/O will timeout. Data integrity is not guaranteed in this condition.
 - Before committing CryptoTarget container or LUN configurations or modifications on an encryption switch or FS8-18 blade, make sure that there are no outstanding zoning transactions in the switch or fabric. If there is an outstanding zoning transaction, the commit operation will fail and result in disabling the LUN. You can check for outstanding zoning transactions by issuing `cfgtransshow` CLI command.
 - LUNs are uniquely identified by the encryption switch or FS8-18 blade using the LUN serial number. The LUN serial number must be unique for LUNs exposed from the same target port. The LUN serial number must be unique for LUNs belonging to different target ports in non-multipathing configurations. Failure to ensure that the serial numbers are unique will result in undefined behavior and may result in faulting the encryption switch or FS8-18 blade.
 - To enable host MPIO, LUNs must also be available through a second target port, hosted on a second encryption switch. The second encryption switch could be in the same fabric, or a different fabric.
 - Hosts should be able to access LUNs through multiple ports for redundancy.
 - For high availability and failover within the fabric, implement an HA cluster of two encryption switches, and host the target port as a virtual target on one of the switches.
 - Don't change the WWN of any node after it has been deployed in an encryption group.
 - To minimize host IO disruption or time-outs during CryptoTarget container failover, it is recommended that the devices (hosts, target ports) are connected to an edge switch in a fabric, and not directly to Encryption switch/blade ports.
 - Always use this two step process when configuring the LUN for encryption, unless the LUN was previously encrypted.
1. Add the LUN as **cleartext** to the CryptoTarget container.
 2. When the LUN comes online and Host I/O starts flowing through the LUN as cleartext, then modify the LUN from cleartext to **encrypt** and **enable_encexistingdata** options to convert the LUN to encryption.

An exception to this LUN configuration process is that if the LUN was previously encrypted by the encryption switch or FS8-18 blade, then the LUN can be added to the CryptoTarget Container with the **-encrypt** and **-lunstate encrypted** options.

Redirection zones

Redirection zones should not be deleted. If a redirection zone is accidentally deleted, I/O traffic cannot be redirected to encryption devices, and encryption is disrupted. To recover, re-enable the existing device configuration by invoking the `cryptocfg --commit` command. If no changes have taken place since the last commit, you should use the `cryptocfg --commit -force` command. This recreates redirection zones related to the device configuration in the zone database, and restores frame redirection, which makes it possible to restore encryption.

To remove access between a given initiator and target, remove both the active zoning information between the initiator and target, and the associated Crypto Target Containers (CTCs). This will remove the associated frame redirection zone information.

Deployment with Admin Domains (AD)

Virtual devices created by the encryption device do not support the AD feature in this release. All virtual devices are part of ADO and AD255. Targets for which virtual targets are created and hosts for which virtual initiators are created must also be in ADO and AD255. If they are not, access from the hosts and targets to the virtual targets and virtual initiators is denied, leading to denial of encryption services.

Master key usage in RKM and SKM environments

In RKM and SKM environments consisting of multiple encryption groups, consider using the same master key for all encryption groups to simplify management.

Do not use DHCP for IP interfaces

Do not use DHCP for either the GbE management interface or the Ge0 and Ge1 interfaces. Assign static IP addresses.

Ensure uniform licensing in HA clusters

Licenses installed on the nodes should allow for identical performance numbers between HA cluster members.

Tape library media changer considerations

In tape libraries where the media changer unit is addressed by a target port that is separate from the actual tape SCSI I/O ports, create a CryptoTarget container for the media changer unit and CryptoTarget containers for the SCSI I/O ports. If a CryptoTarget container is created only for the media changer unit target port, no encryption is performed on this device.

In tape libraries where the media changer unit is addressed by separate LUN at the same target port as the actual tape SCSI I/O LUN, create a CryptoTarget container for the target port, and add both the media changer unit LUN and one or more tape SCSI I/O LUNs to that CryptoTarget container. If only a media changer unit LUN is added to the CryptoTarget container, no encryption is performed on this device.

Turn off host-based encryption

If a host has an encryption capability of any kind, be sure it is turned off before using the encryption engine on the encryption switch or blade. Encryption and decryption at the host may make it impossible to successfully decrypt the data.

Avoid double encryption

Encryption and decryption at tape drives does not affect the encryption switch or blade capabilities, and does not cause problems with decrypting the data. However, double encryption adds the unnecessary need to manage two sets of encryption keys, increases the risk of losing data, may reduce performance, and does not add security.

PID failover

Virtual device PIDs do not persist upon failover within a single fabric HA cluster. Upon failover, the virtual device is assigned a different PID on the standby encryption switch or blade.

Some operating systems view the PID change as an indication of path failure, and will switch over to redundant path in another fabric. In these cases, HA clusters should not be implemented. These operating systems include the following:

- HP-UX prior to 11.x
- All versions of IBM AIX
- Solaris 2.x

Turn off compression on extension switches

If tape pipelining and fast write are enabled on an extension switch, data compression may also be enabled. If data has been encrypted in its path prior to running through the extension switch, data compression should be turned off on the extension switch to increase performance.

Re-keying best practices and policies

Re-keying should be done only when necessary. In key management systems, DEKs are never exposed in an unwrapped or unencrypted state. When using RKM or SKM as the key management system, you must re-key if the master key is compromised. The practice of re-keying should be limited to the following cases:

- Master key compromise in the case of RKM and SKM.
- Insider security breaches.
- As a general security policy as infrequently as every six months or once per year.

When using LKM, DEKs are accessible only to privileged users, and can be compromised only by an insider breach of security.

Manual re-key

Ensure that the link to the key management system is up and running before you attempt a manual re-key.

Latency in re-key operations

Host I/O for regions other than the current re-key region has no latency during a re-key operation. Host I/O for the region where the current re-key is happening has minimal latency (a few milliseconds) because I/O is held until re-key is complete. The I/O sync links (the Ethernet ports labeled Ge0 and Ge1) must be configured, and must both be connected to the I/O sync LAN to enable proper handling of re-key state synchronization in high availability (HA cluster) configurations.

Allow re-key to complete before deleting a container

Do not delete a crypto container while re-key is in session or if re-key is not completed. If you want to delete a container, use the command `cryptocfg --show -rekey --all` to display the status of re-key sessions. If any re-key session is not 100% completed, do not delete the container. If you do delete the container before re-key is complete, and subsequently add the LUN back as cleartext, all data on the LUN is destroyed.

Re-key operations and firmware upgrades

All nodes in an encryption group must be at the same firmware level before starting a re-key or first time encryption operation. Make sure that existing re-key or first time encryption operations complete before upgrading any of the encryption products in the encryption group, and that the upgrade completes before starting a rekey or first time encryption operation.

Do not change LUN configuration while re-keying

Never change the configuration of any LUN that belongs to a Crypto Target Container/LUN configuration while the re-keying process for that LUN is active. If you change the LUN's settings during manual or auto, re-keying or first time encryption, the system reports a warning message stating that the encryption engine is busy and a forced commit is required for the changes to take effect. A forced commit command halts all active re-keying progresses running in all Crypto Target Containers and corrupts any LUN engaged in a re-keying operation. There is no recovery for this type of failure.

Brocade native mode in LKM installations

When using Brocade native mode in LKM installations, manual re-key is highly recommended. If automatic re-key is desired, the key expiry date should be configured only when the LUN is created. Never modify the expiry date after configuring a LUN. If you modify the expiry time after configuring the LUN, the expiration date will not update properly.

Recommendation for Host I/O traffic during online rekeying and first time encryption

You may see failed I/Os if writes are done to a LUN that is undergoing first time encryption or rekeying. It is recommended that host I/O operations are quiesced and not started again until re-key operations or first time encryption operations for the LUN are complete.

Changing IP addresses in encryption groups

Generally, when IP addresses are assigned to the Ge0 and Ge1 ports, they should not be changed. If an encryption group member node IP address is changed, you must de-register the member node from the encryption group, and re-register the member node with the new IP address. If the group leader node IP Address is changed, you must de-register the group leader node from all member nodes, and re-register the group leader node with new IP address on all member nodes. Refer to [Chapter 3, "Encryption configuration using the CLI"](#) for information about how to assign and change cluster interconnect IP addresses.

Disabling the encryption engine

The disable EE interface command `cryptocfg -disableEE [slot no]` should be used only during firmware download, and when the encryption and security capabilities of the encryption engine have been compromised. When disabling the encryption capabilities of the encryption engine, be sure the encryption engine is not hosting any CryptoTarget containers. All Cryptotarget containers hosted on the encryption switch or FS8-18 blade must either be removed from the encryption engine, or be moved to different encryption engine in an HA Cluster or encryption group before disabling the encryption and security capabilities.

Recommendations for Initiator Fan-Ins

For optimal performance at reasonable scaling factors of initiators, targets, and LUNs accessed, Brocade Encryption Engines (EEs) are designed to support a fan-In ratio of between four and eight initiator ports to one target port, in terms of the number of distinct initiator ports to a Crypto Container (i.e., a virtual target port corresponding to the physical target port).

An EE has 6 distinct encryption blocks with 4 ports each port operating at 4Gbps. The architecture of the encryption blocks provides the potential for an aggregate 96 Gbps of full duplex encryption bandwidth, if the performance license is installed. Figure 71 shows the encryption blocks within an EE, and the host initiator to target port fan-ins. Each EE can host up to 256 distinct targets with a mapping of 1024 initiators accessing all the targets. This brings the fan-in ratio for each target to be 1:4 initiators.

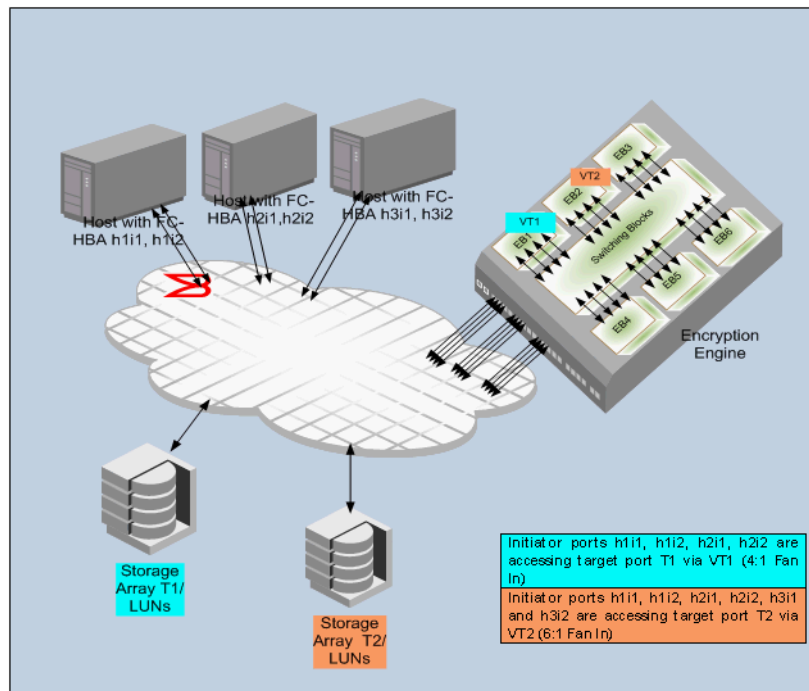


FIGURE 71 Fan-in ratios with performance license installed

The fan-In ratio for a target can be higher depending on the maximum bandwidth accepted by the target. If the I/O throughput across all initiator ports accessing the target port is well balanced, then it is recommended that the maximum fan-In ratio be kept to 8 Initiator ports to 1 target port for optimal performance. Note that this recommendation holds for initiators running at 4Gbps or less. If a mix of 8Gbps and other 4Gbps or less initiator is used then the Maximum Fan-In will depend on the maximum sustained bandwidth these initiators would be pushing together over the link to the same target port and across all the target ports hosted on a given EE.

NOTE

If the performance license is not installed, 48 Gbps of full duplex encryption bandwidth is available of the EE, Each of the six encryption blocks will use two ports instead of four, reducing the fan-in ratio by a factor of two.

Best practices for host clusters in an encryption environment

When host clusters are deployed in an encryption environment, please follow these recommendations:

- If two encryption engines are part of an HA cluster, configure the host/target pair so they have different paths from both encryption engines. Avoid connecting both the host/target pairs to the same encryption engine. This connectivity does not give the full redundancy needed in case of encryption engine failure and failover to another encryption engine in an HA cluster.
- For Windows-based host clusters, when a quorum disk is used, the quorum disk plays a vital role in keeping the cluster synchronized. Please configure the quorum disk to be outside of the encryption environment.

HA Cluster Deployment Considerations and Best Practices

It is mandatory that the two encryption engines in the HA cluster belong to two different nodes for true redundancy. This is always the case for Brocade encryption switches, but is not true if two FS8-18 blades in the same DCX or DCX-4S chassis are configured in the same HA cluster. In Fabric OS version 6.3.0 and later releases, HA cluster creation is blocked when encryption engines belonging to FS8-18 blades in the same DCX or DCX-4S are specified.

Maintenance and Troubleshooting

In this Chapter

- Encryption group and HA cluster maintenance 163
- Troubleshooting examples using the CLI 179
- Management application encryption wizard troubleshooting 181
- Errors related to adding a switch to an existing group 181
- LUN policy troubleshooting 185
- MPIO and internal LUN states. 187

Encryption group and HA cluster maintenance

This section describes advanced configuration options that you can use to modify existing encryption groups and HA clusters, and to recover from problems with one or more member nodes in the group.

All group-wide configuration commands are executed on the group leader. Commands that clear group-related states from an individual node are executed on the node. The commands require Admin or SecurityAdmin permissions.

Removing a node from an encryption group

This procedure permanently removes a node from the encryption group as shown in [Figure 72](#). Upon removal, the HA cluster failover capability and target associations pertaining to the node are no longer present. If you wish to take a node out of a group without disrupting these relationships, use the **cryptocfg --replaceEE** command. Refer to the section “[Replacing an HA cluster member](#)” on page 167 for instructions.

The procedure for removing a node depends on the node’s status within an encryption group. HA cluster membership and Crypto LUN configurations must be cleared before you can permanently remove a member node from an encryption group.

1. Log into the group leader as Admin or SecurityAdmin.
2. If the node is part of an HA cluster, perform the following steps:
 - a. Remove the node from the HA cluster with the **cryptocfg --rem -haclustermember** command.
 - b. Clear all CryptoTarget configurations from the member node with the **cryptocfg --delete -container** command.
3. Determine the state of the node. Log into the member node and enter the **cryptocfg --show -groupmember** command followed by the node WWN. Provide a slot number if the encryption engine is a blade.

6 Encryption group and HA cluster maintenance

```
SecurityAdmin:switch>cryptocfg --show -groupmember \  
10:00:00:05:1e:41:99:bc  
Node Name: 10:00:00:05:1e:41:99:bc (current node)  
State: DEF_NODE_STATE_DISCOVERED  
Role: MemberNode  
IP Address: 10.32.33.145  
Certificate: 10.32.33.145_my_cp_cert.pem  
Current Master Key State: Saved  
Current Master KeyID:  
b8:2a:a2:4f:c8:fd:12:e2:a9:25:d9:5b:58:2c:96:7e  
Alternate Master Key State: Not configured  
Alternate Master KeyID:  
00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00  
  
EE Slot: 0  
SP state: Online  
Current Master KeyID:  
b8:2a:a2:4f:c8:fd:12:e2:a9:25:d9:5b:58:2c:96:7e  
Alternate Master KeyID:  
00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00  
No HA cluster membership
```

- a. If the node is in the DISCOVERED State and the security processor (SP) state is **online** as shown, and you wish to remove the node from the encryption group permanently, proceed to step 4.
- b. If the node is not in the DISCOVERED State, and you wish to remove the node from the encryption group permanently, de-register the node. Log into the group leader and enter the **cryptocfg --dereg -memberrnode** command followed by the node WWN.

```
SecurityAdmin:switch>cryptocfg --dereg -memberrnode 10:00:00:05:1e:41:99:bc  
Operation succeeded.
```

4. Perform one of the following steps to remove the member node from the encryption group.

- a. Log into the member node and enter the **cryptocfg --leave_encryption_group** command. This command clears all node states pertaining to group membership.

```
SecurityAdmin:switch>cryptocfg --leave_encryption_group  
Leave node status: Operation Succeeded.
```

- b. On the group leader, enter the **cryptocfg --eject -memberrnode** command followed by the node WWN. This command removes the node from the encryption group.

```
SecurityAdmin:switch>cryptocfg --eject -memberrnode 10:00:00:05:1e:41:99:bc  
Eject node status: Operation Succeeded.
```

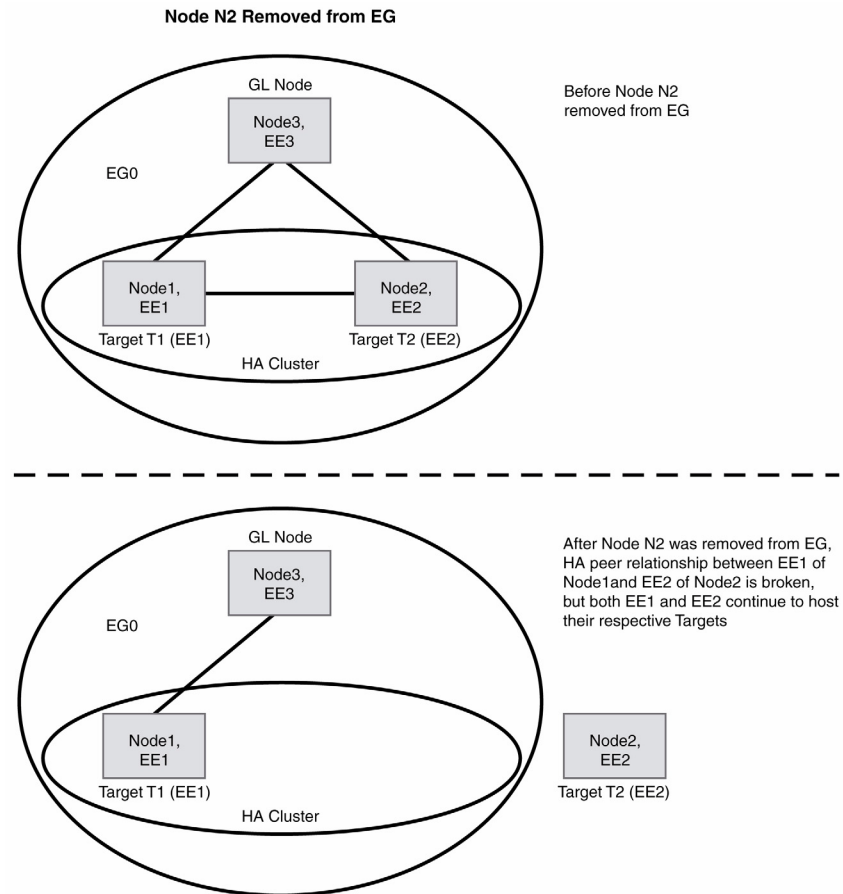


FIGURE 72 Removing a node from an encryption group

Deleting an encryption group

You can delete an encryption group after removing all member nodes following the procedures described in the previous section. The encryption group is deleted on the group leader after you have removed all member nodes.

Before deleting the encryption group, it is highly recommended to remove the group leader from the HA cluster and clear all CryptoTarget and tape pool configurations for the group.

The following example deletes the encryption group “brocade”.

1. Log into the group leader as Admin or SecurityAdmin
2. Enter the **cryptocfg --delete -encgroup** command followed by the encryption group name.

```
SecurityAdmin:switch>cryptocfg --delete -encgroup brocade
Encryption group create status: Operation Succeeded.
```

Removing an HA cluster member

Removing an encryption engine from an HA cluster “breaks” the HA cluster by removing the failover/failback capability for the removed encryption engines. However, the removal of an encryption engine does not affect the relationship between configured containers and the encryption engine that is removed from the HA cluster. The containers still belong to this encryption engine and encryption operations continue.

The remove command should not be used if an encryption engine which failed over exists in the HA Cluster. Refer to the section “[Replacing an HA cluster member](#)” on page 167 for instructions on replacing a failed encryption engine in an HA cluster.

1. Log into the group leader as Admin or SecurityAdmin.
2. Enter the **cryptocfg --remove -haclustermember** command. Specify the HA cluster name and the node WWN to be removed. Provide a slot number if the encryption engine is a blade. The following example removes HA cluster member 10:00:00:05:1e:53:74:87 from the HA cluster HAC2.

```
SecurityAdmin:switch>cryptocfg --remove -haclustermember HAC2 \
10:00:00:05:1e:53:74:87
Remove HA cluster member status: Operation Succeeded.
```

3. Enter **cryptocfg --commit** to commit the transaction.

Displaying the HA cluster configuration

1. Log into the group leader as Admin or SecurityAdmin.
2. Enter the **cryptocfg --show -hacluster -all** command. In the following example, the encryption group brocade has two HA clusters. HAC 1 is committed and has two members. HAC 2 has one member and remains in a defined state until a second member is added and the transaction is committed.

```
SecurityAdmin:switch>cryptocfg --show -hacluster -all
Encryption Group Name: brocade
Number of HA Clusters: 2

HA cluster name: HAC1 - 2 EE entries
Status:          Committed
      WWN              Slot Number  Status
11:22:33:44:55:66:77:00    0          Online
10:00:00:05:1e:53:74:87    3          Online

HA cluster name: HAC2 - 1 EE entry
Status:          Defined
      WWN              Slot Number  Status
10:00:00:05:1e:53:4c:91    0          Online
```


Replacing an HA cluster member

1. Log into the group leader as Admin or SecurityAdmin.
2. Enter the **cryptocfg --replace -haclustermember** command. Specify the HA cluster name, the node WWN of the encryption engine to be replaced, and the node WWN of the replacement encryption engine. Provide a slot number if the encryption engine is a blade. The replacement encryption engine must be part of the same encryption group as the encryption engine that is replaced.

```
SecurityAdmin:switch>cryptocfg --replace -haclustermember HAC2 \
10:00:00:05:1e:53:4c:91 10:00:00:05:1e:39:53:67
Replace HA cluster member status: Operation Succeeded.
```

3. Enter **cryptocfg --commit** to commit the transaction.

Case 1: Replacing a failed encryption engine in an HA cluster

Assume a working HA cluster with two operational encryption engines, EE1 and EE2. The target T1 is hosted on EE1 and target T2 is hosted on EE2. Refer to [Figure 73](#).

EE2 fails and generates an offline notification. The target hosted on EE2 (T2 in this case) automatically fails over to EE1. Even though the target T2 is now hosted on EE1 because of the failover process, the target association is still EE2, and the container status is displayed on the hosting node as failover. Use the **cryptocfg --show -container crypto target container name -stat** command to display the container status.

1. Invoke the **cryptocfg --replace -haclustermember** command on the group leader to replace the failed encryption engine (EE2) with another encryption engine (EE3). This operation effectively removes the failed encryption engine (EE2) from the HA cluster and adds the replacement encryption engine (EE3) to the HA cluster. The target associations (T2) from the failed encryption engine (EE2) are transferred to the replacement encryption engine (EE3).
2. Commit the transaction. If failback mode is set to **auto**, the target (T2) which failed over earlier to EE1 automatically fails back to the replaced encryption engine (EE3).
3. Once the transaction is committed, remove the failed encryption engine from the encryption group.

6 Encryption group and HA cluster maintenance

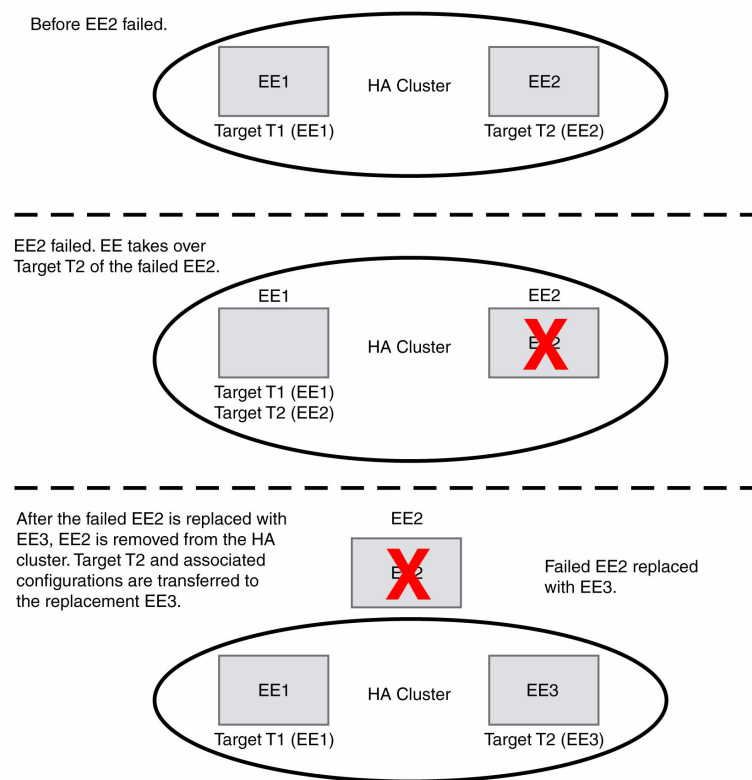


FIGURE 73 Replacing a failed encryption engine in an HA cluster

Case 2: Replacing a “live” encryption engine in an HA cluster

1. Invoke the `cryptocfg --replace -haclustermember` command on the group leader to replace the live encryption engine EE2 with another encryption engine (EE3). This operation effectively removes EE2 from the HA cluster and adds the replacement encryption engine (EE3) to the HA cluster. The target associations (T2) from the replaced encryption engine (EE2) are transferred to the replacement encryption engine (EE3).
2. Commit the transaction.
3. Remove the encryption engine EE2 from the encryption group.

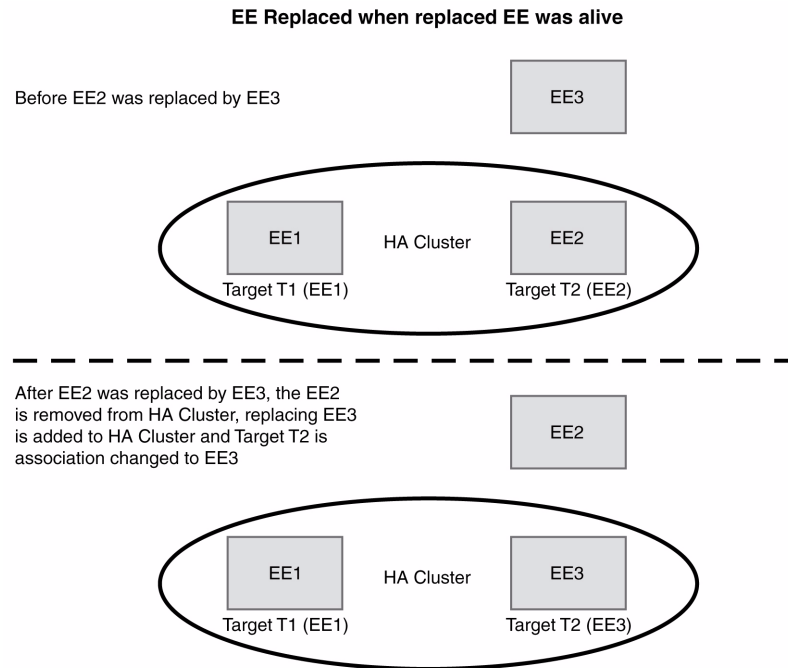


FIGURE 74 Replacing a “live” encryption engine in an HA cluster.

Deleting an HA cluster member

This command dissolves the HA cluster and removes failover capability from the participating encryption engines.

1. Log into the group leader as Admin or SecurityAdmin.
2. Enter the **cryptocfg --delete -hacluster** command. Specify the name of the HA cluster you wish to delete.

```
SecurityAdmin:switch>cryptocfg --delete -hacluster HAC1
Delete HA cluster status: Operation succeeded.
```

3. Enter the **cryptocfg --commit** command to commit the transaction.

Performing a manual failback of an encryption engine

By default, failback occurs automatically if an encryption engine that failed was replaced or comes back online. When **manual failback** policy is set in the encryption group, you must invoke a manual failback of the encryption engine after the failing encryption engine was restored or replaced. Failback includes all of the encryption engine's target associations. Failback returns all encryption operations to the original encryption engine after it has been restored, or it transfers operations to a replacement encryption engine if the original encryption engine was replaced. The failback operation can only be performed within an HA cluster.

1. Log into the group leader as Admin or SecurityAdmin.
2. Enter the **cryptocfg --failback -EE** command. Specify the node WWN of the encryption engine to which failover occurred earlier and which is now performing all encryption tasks (current encryption engine), followed by the node WWN of the encryption engine to which failback should occur ("new" encryption engine). Specify a slot number if the encryption engine is a blade.

```
SecurityAdmin:switch>cryptocfg --failback -EE 10:00:00:05:1e:53:4c:91 \
10:00:00:05:1e:39:53:67
Operation Succeeded
```

Failover/failback example

The following example illustrates the states associated with the encryption engines during an active failover and failback process.

- EE2 fails over to EE1.

```
SecurityAdmin:switch>cryptocfg --show -hacluster -all
Encryption Group Name: brocade
Number of HA Clusters: 1

HA cluster name: HAC3- 2 EE entries
Status:          Committed

      WWN                Slot Number  Status
EE1 => 10:00:00:05:1e:53:89:dd      0      Online - Failover active
EE2 => 10:00:00:05:1e:53:fc:8a      0      Offline
```

- The failed EE2 has come back online, Failover is still active:

```
SecurityAdmin:switch>cryptocfg --show -hacluster -all
Encryption Group Name: brocade
Number of HA Clusters: 1

HA cluster name: HAC3 - 2 EE entries
Status:          Committed

      WWN                Slot Number  Status
EE1 => 10:00:00:05:1e:53:89:dd      0      Online - Failover active
EE2 => 10:00:00:05:1e:53:fc:8a      0      Online
```

- A manual failback is issued.

```
SecurityAdmin:switch>cryptocfg --failback -EE 10:00:00:05:1e:53:89:dd 0 \
10:00:00:05:1e:53:fc:8a 0
Operation succeeded.
```

- After the failback completes, the `-cryptocfg --show -hacluster -all` command no longer reports active failover.

```
SecurityAdmin:switch>cryptocfg --show -hacluster -all
Encryption Group Name: brocade_1
Number of HA Clusters: 1

HA cluster name: HAC3 - 2 EE entries
Status:          Committed

      WWN                Slot Number  Status
EE1 => 10:00:00:05:1e:53:89:dd      0      Online
EE2 => 10:00:00:05:1e:53:fc:8a      0      Online
```

Encryption group merge and split use cases

This section describes recovery scenarios for the following cases:

- [“A member node failed and is replaced”](#) on page 171
- [“A member node reboots and comes back up”](#) on page 173
- [“A member node lost connection to the group leader”](#) on page 173
- [“A member node lost connection to all other nodes in the encryption group”](#) on page 174
- [“Several member nodes split off from an encryption group”](#) on page 174

A member node failed and is replaced

Assumptions

N1, N2 and N3 form an encryption group and N2 is the group leader node. N3 and N1 are part of an HA cluster. Assume that N3 failed and you want to replace the failed N3 node with an alternate node N4.

Impact

When N3 failed, all devices hosted on the encryption engines of this node failed over to the peer encryption engines in N1, and N1 now performs all of the failed node's encryption services. Re-key sessions owned by the failed encryption engine are failed over to N1.

Recovery

1. Configure the IP address of the new node that is replacing the failed node, and the IP addresses of the I/O cluster sync ports (Ge0 and Ge1), and initialize the node with the **cryptocfg --initnode** command. Refer to [“I/O sync link configuration”](#) on page 88 and [“Encryption switch initialization”](#) on page 90.
2. Register the new node IP address and CP certificate with the group leader node. Refer to the section [“Basic encryption group configuration”](#) on page 95 for instructions.
3. On the group leader node, export the member node certificate. Refer to [“Exporting a certificate”](#) on page 93 for instructions.
4. On the group leader node, import the member node certificate. Refer to [“Importing a certificate”](#) on page 94 for instructions.
5. On the group leader node, register the member node with the group leader node. Enter the **cryptocfg --reg -membernode** command with appropriate parameters to register the member node. Specify the member node’s WWN, Certificate filename, and IP address when executing this command. Successful execution of this command distributes all necessary node authentication data to the other members of the group.

```
SecurityAdmin:switch>cryptocfg --reg -membernode \  
10:00:00:05:1e:39:14:00 enc_switch1_cert.pem 10.32.244.60  
Operation succeeded.
```

6. Add the new node to the encryption group by invoking the **cryptocfg --add -membernode** command on the group leader node. Provide the node WWN and a slot number if the encryption engine is a blade.

```
SecurityAdmin:switch>cryptocfg --add -membernode 10:00:00:05:1e:39:14:00  
Add node status: Operation Succeeded.
```
7. Initialize and enable the encryption engines. On the new node, invoke the **cryptocfg --initEE** and **cryptocfg --regEE** commands to initialize the encryption engines.
8. For RKM and SKM environments, register the new node with the key appliance.
9. On the new node, invoke **cryptocfg -initEE** and **cryptocfg -regEE** to initialize the Encryption Engines, and do **reboot -f** an encryption switch, or **SlotPowerOff** and **SlotPowerOn** for an FS8-18 blade in a DCX or DCX-4S.
10. After the new node has come online, invoke the **cryptocfg --enableEE [slot_number]** command to enable crypto operations on the node’s encryption engines.
11. Replace the failed encryption engine on N3 with the encryption engine of the new node N4 to restore broken HA cluster peer relationships. Use the **cryptocfg --replaceEE** command. Refer to the section [“Replacing an HA cluster member”](#) on page 167 for instructions.
12. Remove the failed node from the encryption group. Follow the procedures described in the section [“Removing a node from an encryption group”](#) on page 163.

A member node reboots and comes back up

Assumptions

N1, N2 and N3 form an encryption group and N2 is the group leader node. N3 and N1 are part of an HA cluster. Assume that N3 reboots and comes back up.

Impact

When N3 reboots, all devices hosted on the encryption engines of this node automatically fail over to the peer encryption engine N1, and N1 now performs all of the rebooted node's encryption services. Any re-key sessions in progress continue. Re-key sessions owned by N3's encryption engine are failed over to N1.

Recovery

If **auto failback** policy is set, no intervention is required. After the node has come back up, all devices and associated configurations and services that failed over earlier to N1 fail back to N3. The node resumes its normal function.

If **auto failback** policy is not set, invoke a manual failback if required. Refer to the section [“Performing a manual failback of an encryption engine”](#) on page 170 for instructions.

A member node lost connection to the group leader

Assumptions

N1, N2 and N3 form an encryption group, and N2 is the group leader node. N3 and N1 are part of an HA cluster. Assume that N3 lost connection to the group leader node N2 but still maintains communications with other nodes in the encryption group.

Impact

Failover to N1 does not occur, because the isolated node and the encryption engines' encryption services continue to function normally. However the disconnect of N3 from the group leader breaks the HA cluster and failover capability between N3 and N1.

You cannot configure any CryptoTargets, LUN policies, tape pools, or security parameters that would require communication with the isolated member node. In addition, you cannot start any re-key operations (auto or manual).

Refer to the section [“Configuration impact of encryption group split or node isolation”](#) on page 176 for more information on which configuration changes are allowed.

Recovery

Restore connectivity between the isolated node and the group leader. No further intervention is required.

A member node lost connection to all other nodes in the encryption group

Assumptions

N1, N2 and N3 form an encryption group and N2 is the group leader node. N3 and N1 are part of an HA cluster. Assume that N3 lost connection with all other nodes in the group. Node N3 finds itself isolated from the encryption group and, following the group leader succession protocol, elects itself as group leader. This action splits the encryption group into two encryption group islands. EG1 includes the original encryption group minus the member node N3 that lost connection to the encryption group. EG2 consists of a single node N3, which functions as the group leader.

Impact

- The two encryption group islands keep functioning independently of each other as far as host I/O encryption traffic is concerned.
- Each encryption group registers the missing members as “offline”.
- The isolation of N3 from the group leader breaks the HA cluster and failover capability between N3 and N1.
- You cannot configure any CryptoTargets, LUN policies, tape pools, or security parameters on any of the group leaders. This would require communication with the “offline” member nodes. You cannot start any re-key operations (auto or manual) on any of the nodes. Refer to the section [“Configuration impact of encryption group split or node isolation”](#) on page 176 for more information on which configuration changes are allowed.

Recovery

1. Restore connectivity between the two separate encryption group islands.

When the lost connection is restored, an automatic split recovery process begins. The current group leader and the former group leader (N3 and N2 in this example) arbitrate the recovery, and the group leader with the majority number of members (N2) becomes group leader. If the number of member nodes is the same, the group leader node with the highest WWN becomes group leader.

2. After the encryption group enters the **converged** state, execute the `cryptocfg --commit` command on the group leader node to distribute the crypto-device configuration from the group leader to all member nodes.

Should you decide to remove the isolated node N3, follow the procedures described in the section [“Removing a node from an encryption group”](#) on page 163.

Several member nodes split off from an encryption group

Assumptions

N1, N2, N3, and N4 form an encryption group and N2 is the group leader node. N3 and N1 are part of an HA cluster. Assume that both N3 and N4 lost connection with the encryption group but can still communicate with each other. Following the group leader succession protocol, N3 elects itself as group leader to form a second encryption group with itself and N4 as group members. We now have two encryption groups, EG1 (group leader N2 + N1), and EG2 (group leader N3 + N4).

Impact

- The two encryption groups continue to function independently of each other as far as host I/O encryption traffic is concerned.

- Each encryption group registers the missing members as “offline”.
- The isolation of N3 from the original encryption group breaks the HA cluster and failover capability between N3 and N1.
- You cannot configure any CryptoTargets, LUN policies, tape pools, or security parameters on any of the group leaders. This would require communication with the “offline” member nodes. You cannot start any re-key operations (auto or manual) on any of the nodes. Refer to the section [“Configuration impact of encryption group split or node isolation”](#) on page 176 for more information on which configuration changes are allowed.

Recovery

1. Restore the connection between the nodes in the separate encryption group islands, that is, between nodes N3, N4 and between nodes N1 and N2.

When the lost connection is restored, an automatic split recovery process begins. The two group leaders (N3 and N2 in this example) arbitrate the recovery, and the group leader node with the highest WWN becomes group leader. If the number of nodes in each group is not equal, the group leader for the group with the largest number of members becomes group leader.

2. After the encryption group enters the **converged** state, execute the **cryptocfg --commit** command on the group leader node to distribute the crypto-device configuration from the group leader to all member nodes.

Configuration impact of encryption group split or node isolation

When a node is isolated from the encryption group or the encryption group is split to form separate encryption group islands, the defined or registered node list in the encryption group is not equal to the current active node list, and the encryption group is in a DEGRADED state rather than in a CONVERGED state. [Table 9](#) and [Table 10](#) list configuration changes that are allowed and disallowed under such conditions.

TABLE 9 Allowed Configuration Changes

Configuration Type	Allowed configuration changes
Encryption group	<ul style="list-style-type: none"> • Adding a node to the encryption group • Removing a node from the encryption group • Invoking a node leave command • Deleting an encryption group • Registering a member node (IP address, certificates)
HA cluster	<ul style="list-style-type: none"> • Removing an encryption engine from an HA cluster • Deleting an HA cluster
Security & key vault	<ul style="list-style-type: none"> • Initializing a node • Initializing an encryption engine • Re-registering an encryption engine • Zeroizing an encryption engine

TABLE 10 Disallowed Configuration Changes

Configuration Type	Disallowed configuration changes
Security & key vault	<ul style="list-style-type: none"> • Register or modify key vault settings • Generating a master key • Exporting a master key • Restoring a master key • Enabling or disabling encryption on an encryption engine
HA cluster	<ul style="list-style-type: none"> • Creating an HA cluster • Adding an encryption engine to an HA cluster • Modifying the failback mode
Crypto Device (target/LUN/tape)	<ul style="list-style-type: none"> • Creating a CryptoTarget container • Adding initiators or LUNs to a CryptoTarget container • Removing initiators or LUNs from a CryptoTarget container • Modifying LUNs or LUN policies • Creating or deleting a tape pool • Modifying a tape pool policy • Starting a manual re-keying session • Performing a manual failback of containers • Deleting a CryptoTarget container

General encryption troubleshooting using the CLI

Table 11 lists the commands you can use to check the health of your encryption setup. Table 12 provides additional information for failures you might encounter while configuring switches using the CLI.

TABLE 11 General troubleshooting tips using the CLI

Command	Activity
<code>supportsave</code>	Check whole system configuration. Run RAS logs. Run RAS traces. Run Security Processor (SP) logs (mainly <code>kpd.log</code>).
<code>errdumpall</code> <ul style="list-style-type: none"> • <code>errdumpall grep KAC</code> • <code>errdumpall grep CVLM</code> • <code>errdumpall grep CVLC</code> • <code>errdumpall grep SPM</code> • <code>errdumpall grep CNM</code> 	Run error logs: <ul style="list-style-type: none"> • Key Adapter Module error logs. • Crypto Virtual device module error logs. • Crypto LUN module error logs. • SP Manager error logs. • Cluster Node Manager error logs. <p>NOTE: <code>errdumpall</code> requires root user access.</p>
<code>configshow</code>	Check whole system persistent configuration database dump. Check for SPM-, CVLM-, and CNM-related persistent database entries.
<code>cfgshow</code>	Check for redirection zones starting with "red_XXX" in defined database for virtual and physical devices.
<code>nsshow</code>	Check for crypto virtual target and crypto virtual initiator entries for VT/VI
<code>switch:SecurityAdmin> cryptocfg --show -groupcfg</code>	Check key vault connection status. Check encryption group/cluster status. Note: CONVERGED status means the cluster is formed successfully.
<code>switch:SecurityAdmin> cryptocfg --show -groupmember -all</code>	<ol style="list-style-type: none"> 1 Check encryption group/cluster member status. Note: DISCOVERED state means the member is currently part of a cluster. 2 Check encryption engine/SP and KEK status. Note: SP state ONLINE means encryption engine is enabled for encryption with valid KEK (Link Key or Master Key).

TABLE 12 General errors related to using the CLI

Problem	Resolution
When the connectivity to an LKM key vault is lost, a RAS log message is not generated.	Issue any of the <code>cryptocfg</code> commands that attempt a key vault communication (such as the <code>cryptocfg --show -groupcfg</code> command).
After you create an encryption group using RKM, a newly created container's LUN state changes between "Write metadata is pending" and "Write metadata is in progress" with continuous [RKD-1001] messages displayed on the console.	Power cycle the DCX chassis and then issue the <code>cryptocfg --enableEE [slot number]</code> command to bring the container's LUN state to Encryption Enabled. If the eth0 IP address on the Brocade Encryption Switch or on the FS8-18 port blade has been modified, a reboot is required.
LUN state for some LUNS remains in "initialize" state on the passive path.	This is expected behavior. The LUNs exposed through Passive paths of the target array will be in either Initialize or LUN Discovery Complete state so long as the paths remain in passive condition. When the passive path becomes active, the LUN changes to Encryption Enabled. Use the <code>--show -LUN</code> command with the <code>-stat</code> option to check the LUN state.

6 General encryption troubleshooting using the CLI

TABLE 12 General errors related to using the CLI (Continued)

Problem	Resolution
<p>A backup fails because the LUN is always in the initialize state for the tape container.</p> <p>Tape media is encrypted and gets a key which is archived in the key vault. The key is encrypted with a master key. At a later point in time you generate a new master key. You decide to use this tape media to back up other data. You rewind the tape, erase the tape, relabel the tape, and start a backup from the start of the tape. When the first command comes from the host, the key vault is queried for the tape media based on the media serial number. Since this tape media was used previously, the key is already present in the key vault for this media serial number but this key is encrypted with the old master key and that master key is not present in the switch. You cannot create a new key for this tape media because, per policy, there can be only one key per media.</p>	<p>Use one of two resolutions:</p> <ul style="list-style-type: none"> • Load the old master key on the switch at an alternate location. The key for the tape media can then be decrypted. • Delete the key for the tape media from the key vault. This forces the switch to create a new key for the tape media. <p>Until you start the backup, the LUN remains in “initialize” state.</p>
<p>“Invalid certificate” error message received when doing a KAC certificate exchange between the Brocade Encryption Switch and a key management system appliance. This error is due to the Brocade Encryption Switch time being ahead of the appliance time.</p>	<p>Use one of two resolutions:</p> <ul style="list-style-type: none"> • Change the appliance time to match the start period of the KAC certificate. • Change the Brocade Encryption Switch time to synchronize with the appliance time. <p>Upon completion, regenerate the KAC certificate and then do another KAC certificate exchange with the appliance.</p>
<p>“Temporarily out of resources” message received during re-key or first time encryption.</p>	<p>Re-key or first time encryption sessions are pending due to resource unavailability. A maximum of twelve sessions including rekey (manual or auto) and first time encryption sessions are supported per encryption switch or blade and two sessions per target. The system checks once every hour to determine, if there are any re-key or first time encryption sessions pending. If resources are available, the next session in the queue is processed. There may be up to an hour lag before the next session in the queue is processed. It is therefore recommended that you do not schedule more than 12 re-key or first time encryption sessions.</p>
<p>HA cluster creation fails with error, Create HA cluster status: The IO link IP address of the EE (online) is not configured, even though both the addresses are set and accessible.</p>	<p>The IP addresses for the I/O link ports should be configured before enabling the EE. Failure to do so results in unsuccessful HA Cluster creation. If the IP addresses for these ports were configured after the EE is enabled, reboot the encryption switch or slotpoweroff/slotpoweron the encryption blade to sync up the IP address information to the EE.</p>
<p>Re-keying fails with error “Disabled (Key not in sync)”.</p>	<p>Re-keying was started on a remote EE but the local EE is not capable of starting re-key because the key returned from key vault does not match with the Key ID used by remote EE. You will need to re-enable the LUN after the keys are synced between two key vaults properly using the needs to cryptocfg -discoverLUN <Container Name> command.</p>
<p>cryptocfg --commit fails with message “Default zone set to all access at one of nodes in EG.”</p>	<p>Default zoning must be set to no access. Refer to “Setting default zoning to no access” on page 87.</p>

Troubleshooting examples using the CLI

Encryption Enabled Crypto Target LUN

The LUN state should be **Encryption enabled** for the host to see the Crypto LUN.

```
switch:FabricAdmin> cryptocfg --show -LUN kmfvt 0 21:01:00:e0:8b:a9:ac:d2 -stat
Container name:      lkmfvt
Type:               disk
EE node:            10:00:00:05:1e:41:9a:88
EE slot:            0
Target:             50:06:01:60:10:60:06:3a 50:06:01:60:90:60:06:3a
Target PID:         030700
VT:                 20:00:00:05:1e:41:4d:79 20:01:00:05:1e:41:4d:79
VT PID:             012401
Host:                21:01:00:e0:8b:a9:ac:d2 20:01:00:e0:8b:a9:ac:d2
Host PID:           030300
VI:                  20:02:-00:05:1e:41:4d:79 20:03:00:05:1e:41:4d:79
VI PID:             012402
LUN number:         0x0
LUN type:           disk
LUN serial number: 600601604F0B0900847C800FCE0FDD11000000000000000000000E000000
                   000000
Encryption mode:    encrypt
Encryption format:  native
Encrypt existing
data:               disabled
Rekey:              disabled
LUN state:         Encryption enabled
Encryption
algorithm:          AES256-XTS
Key ID state:       Read write
Key ID:             c5:b7:d3:04:53:4b:f8:19:7d:46:87:a7:04:42:68:88
Key creation time:  Thu Jun 26 19:28:27 2008
Operation succeeded
```

Encryption Disabled Crypto Target LUN

If the LUN state is **Encryption Disabled** the host will not be able to access the Crypto LUN.

```
switch: FabricAdmin>> cryptocfg --show -LUN kmfvt 0 21:01:00:e0:8b:a9:ac:d2 -stat
Container name:          lkmfvt
Type:                   disk
EE node:                10:00:00:05:1e:43:fe:00
EE slot:                4
Target:                 50:06:01:61:10:60:06:3a 50:06:01:60:90:60:06:3a
Target PID:             890c00
VT:                    20:04:00:05:1e:41:4d:79 20:05:00:05:1e:41:4d:79
VT PID:                 01b201
Host:                   21:00:00:e0:8b:89:ac:d2 20:00:00:e0:8b:a9:ac:d2
Host PID:               890800
VI:                    20:06:-00:05:1e:41:4d:79 20:07:00:05:1e:41:4d:79
VI PID:                 01b202
LUN number:             0x1
LUN type:                disk
LUN serial number:      600601604F0B0900857C800FCE0FDD1100000000000000000000000000000000
                        F000000000000000
Encryption mode:        encrypt
Encryption format:      native
Encrypt existing data:  disabled
Rekey:                  disabled
LUN state:             Disabled (Unable to retrieve key by key ID found from metadata)
Encryption algorithm:   AES256-XTS
Key ID state:           Read write
Key ID:                 77:93:09:a1:eb:00:af:55:ef:8f:a3:53:e7:a5:9d:ef
Key creation time:      Thu Jun 26 19:28:27 2008
Operation succeeded
```

Management application encryption wizard troubleshooting

- [Errors related to adding a switch to an existing group](#) 181
- [Errors related to adding a switch to a new group](#) 182
- [General errors related to the Configure Switch Encryption wizard](#) 184

Errors related to adding a switch to an existing group

[Table 13](#) lists configuration task errors you might encounter while adding a switch to an existing group, and describes how to troubleshoot them.

TABLE 13 Error recovery instructions for adding a switch to an existing group

Configuration task	Error description	Instructions
Initialize the switch	Unable to add switch to encryption group. The switch is no longer a group leader or does not contain a group.	<p>Manual option:</p> <p>To add a switch to the group on the leader switch:</p> <ol style="list-style-type: none"> 1 Re-launch the Configure Switch Encryption wizard and create a new encryption group on the leader switch. 2 When that group is created, launch the Configure Switch Encryption wizard again and add the switch to the group.
Initialize the switch	The switch was not properly initialized and was aborted because it is unavailable.	Re-run the Configure Switch Encryption wizard for the switch.
Add the switch to the encryption group	Adding the switch to the encryption group failed.	Re-run the Configure Switch Encryption wizard for the switch.
Enable the encryption engines	A failure occurred while attempting to enable encryption engines on the switch.	<ol style="list-style-type: none"> 1 Remove the switch from the group using the Group Members tab on the Encryption Group Properties dialog box. 2 Re-run the Configure Switch Encryption wizard for the switch. <p>Manual Option:</p> <ol style="list-style-type: none"> 1 Save the switch's public key certificate to a file using the Switch Encryption Properties dialog box. 2 Follow the Key Vault instructions for RSA/Decru/Other key vault.
Save the switch's public key certificate to a file.	<p>The switch's public key certificate could not be saved to a file.</p> <p>Note: Verify that the path name and the file name that you are using are both valid and that you have write permissions for the file.</p>	<ol style="list-style-type: none"> 1 Remove the switch from the group using the Group Members tab on the Encryption Group Properties dialog box. 2 Re-run the Configure Switch Encryption wizard for the switch. <p>Manual Option:</p> <ol style="list-style-type: none"> 1 Save the switch's public key certificate to a file using the Switch Encryption Properties dialog box. 2 Follow the Key Vault instructions.

Errors related to adding a switch to a new group

Table 14 lists configuration task errors you might encounter while adding a switch to a new group, and describes how to troubleshoot them.

TABLE 14 Error recovery instructions for adding a switch to a new group

Configuration task	Error description	Instructions
Initialize the switch	Unable to initialize the switch due to an error response from the switch.	Diagnose the problem using standard switch CLI commands.
	The switch was not properly initialized and was aborted because it is unavailable.	Re-run the Configure Switch Encryption wizard for the switch.
Create encryption group on the switch	A failure occurred while attempting to create a new encryption group on the switch.	<ol style="list-style-type: none"> 1 Click the Refresh button on the Configure Switch Encryption dialog box to synchronize the data and the database. 2 Re-run the Configure Switch Encryption wizard for the switch.
Register one or more key vaults	A failure occurred while attempting to register one or more key vaults for a group on the switch.	<ol style="list-style-type: none"> 1 Remove the switch from the group using the Group Members tab on the Encryption Group Properties dialog box. 2 Re-run the Configure Switch Encryption wizard for the switch. <p>Manual Option:</p> <ol style="list-style-type: none"> 1 Launch the Encryption Group Properties dialog box and click the General tab. 2 From the General dialog box, click Load from File to install key vault certificates, and then click OK to save the information on to the switch. 3 Follow the Key Vault instructions.
Enable the encryption engines	A failure occurred while attempting to enable encryption engines on the switch.	<ol style="list-style-type: none"> 1 Remove the switch from the group using the Group Members tab on the Encryption Group Properties dialog box. 2 Re-run the Configure Switch Encryption wizard for the switch. <p>Manual Option:</p> <ol style="list-style-type: none"> 1 Launch the Switch Encryption Properties dialog box. 2 Save the switch's public key certificate to a file using the Switch Encryption Properties dialog box. 3 Follow the Key Vault instructions for the key vault.

TABLE 14 Error recovery instructions for adding a switch to a new group (Continued)

Configuration task	Error description	Instructions
Create a new master key (if the key vault type is not NetApp)	A failure occurred while attempting to create a new master key.	<ol style="list-style-type: none"> 1 Remove the switch from the group using the Group Members tab on the Encryption Group Properties dialog box. 2 Re-run the Configure Switch Encryption wizard for the switch. <p>Manual Option:</p> <ol style="list-style-type: none"> 1 Launch the Encryption Group Properties dialog box, and click Security. 2 Click the Master Key Action button and select Create New Master Key to generate a new master key.
Save the switch's public key certificate to a file.	<p>The switch's public key certificate could not be saved to a file.</p> <p>Note: Verify that the path name and the file name that you are using are both valid and that you have write permissions for the file.</p>	<ol style="list-style-type: none"> 1 Remove the switch from the group using the Group Members tab on the Encryption Group Properties dialog box. 2 Re-run the Configure Switch Encryption wizard for the switch. <p>Manual Option:</p> <ol style="list-style-type: none"> 1 Save the switch's public key certificate to a file using the Switch Encryption Properties dialog box. 2 Follow the Key Vault instructions

General errors related to the Configure Switch Encryption wizard

Table 15 provides additional information for failures you might encounter while configuring switches using the Configure Switch Encryption wizard.

TABLE 15 General errors related to the Configure Switch Encryption wizard

Problem	Resolution
Initialization fails on the encryption engine after the encryption engine is zeroized.	Reboot the switch.
Configuration Commit fails with message “Default zone set to all access at one of nodes in EG.”	Default zoning must be set to no access. Refer to “Setting default zoning to no access” on page 87.

LUN policy troubleshooting

Table 16 may be used as an aid in troubleshooting problems related to LUN policies.

TABLE 16 LUN policy troubleshooting

Case	Reasons for the LUN getting disabled by the encryption switch	Action taken	If you do not need to save the data:	If you need to save the data:
1	The LUN was modified from encrypt policy to cleartext policy but metadata exists.	LUN is disabled. Reason code: Metadata exists but the LUN policy is cleartext.	Issue the cryptocfg -enable -LUN command on one path of the LUN. This erases the metadata on the LUN and the LUN is then enabled with cleartext policy. Issue the cryptocfg -discoverLUN command on other paths of the LUN in the DEK cluster to enable the LUN.	Modify the LUN back to encrypt policy.
2	The LUN was set up with an encrypt policy and the LUN was encrypted (metadata is present on the LUN), but the DEK for the key ID present in the metadata does not exist in the key vault.	LUN is disabled. Reason code: Metadata exists but the DEK for the key ID from the metadata does not exist.	Modify the LUN policy to cleartext. The subsequent handling is same as in case 1.	Make sure the key vault has the DEK and when the DEK gets restored to the key vault, perform one of the following tasks on one of the paths of the LUN to enable the LUN: <ul style="list-style-type: none"> • Issue the cryptocfg -discoverLUN command • Remove the LUN from the container and then add it back • Bounce the target port Then issue the cryptocfg -discoverLUN command on other paths of the LUN in the DEK cluster.
3	The LUN was set up with an encrypt policy and the LUN was encrypted (metadata is present on the LUN), but the current state of the LUN is cleartext instead of encrypted.	LUN is disabled. Reason code: Metadata exists, but the LUN policy is indicated as cleartext.	Modify the LUN policy to cleartext. The subsequent handling is the same as in case 1.	Remove the LUN from the container and then add the LUN back with the LUN state as encrypted, or issue the cryptocfg -enable -LUN command on one of the paths of the LUN which will enable the LUN by using the appropriate key. Then issue the cryptocfg -discoverLUN command on other paths of the LUN in the DEK cluster to enable the LUN.

Loss of encryption group leader after power outage

When all nodes in an encryption group, HA Cluster, or DEK Cluster are powered down due to catastrophic disaster or power outage to whole data center, and the group leader node either fails to come back up when the other nodes are powered on, or the group leader is kept powered down, the member nodes lose information and knowledge about the encryption group. If this happens, no crypto operations or commands (except node initialization) are available on the member node after the power-cycle. This condition persists until the group leader back is online.

When a group leader node fails to come back up, the group leader node can be replaced. You can do this in one of two ways:

- Promote an existing member node to group leader.
- Replace the failed group leader node with a new node.

Use the following procedure to make one of existing member nodes into a group leader node, and make the encryption group functional again:

1. On one of the member nodes, create the encryption group with same encryption group name. That node then becomes the group leader node, and the related configurations are kept intact for the encryption group.
2. For any containers hosted on the failed group leader node, issue the **cryptocfg - -replace** command to change the WWN association of containers from failed group leader node to the new group leader node for all containers on the encryption engine.

Use the following procedure to replace the failed group leader node with a new node:

1. On the new node, perform the switch/node initialization steps as described in Chapter 3.
2. Create an encryption group on the new node with the same encryption group name as before.
3. Use the **configdownload** command to download previously uploaded group leader node and encryption group configuration files to the new node.
4. For any containers hosted on the failed group leader node, issue the **cryptocfg - -replace** command to change the WWN association of containers from failed group leader node to the new group leader node for all containers on the encryption engine.

MPIO and internal LUN states

The Internal LUN State field displayed within the **cryptocfg -show -LUN** command output does not indicate the host-to-storage path status for the displayed LUN, but rather the internal LUN state as known by the given encryption engine. Due to the transparent and embedded nature of this encryption solution, the host-to-storage array LUN path status can only be displayed by using host MPIO software.

For example, assume there are two paths from a host through two encryption switches to a LUN configured within an active/passive storage array. If the LUN is trespassed, and the active and passive paths to the LUN are swapped, the host MPIO software will continue to indicate that only one path is active to the LUN, but the Brocade encryption switch internal LUN states for both paths will now likely be displayed as Encryption Enabled.

In active/passive storage array environments, for troubleshooting purposes, you may want to update the encryption engine Internal LUN States to match those of their host MPIO path states. You can do this by running the **cryptocfg -discoverLUN <crypto target container name>** command for the encryption engines that own paths to the LUN in question. This command forces a LUN discovery, causing the encryption engine's Internal LUN State to be updated.

Suspension and resumption of re-keying operations

A re-key may be suspended or fail to start for several reasons:

- The LUN goes offline or the encryption switch fails and reboots. Re-key operations are resumed automatically when the target comes back online or the switch comes back up. You cannot abort an in-progress re-key operation.
- An unrecoverable error is encountered on the LUN and the in-progress re-key operation halts. The following LUN errors are considered unrecoverable:


```
SenseKey: 0x3 - Medium Error.
SenseKey: 0x4 - Hardware Error.
SenseKey: 0x7 - Data Protect.
```
- An unrecoverable error is encountered during the re-key initialization phase. The re-key operation does not begin and a CRITICAL error is logged. All host I/O comes to a halt. All cluster members are notified.
- For any unrecoverable errors that may occur during any other phase of the process, the re-key operation is suspended at that point and a CRITICAL error is logged. All cluster members are notified. Host I/O to all regions of the LUN is halted. Only READ operations are supported for the scratch space region of the LUN used for storing the status block of the re-key operation.

Once all errors have been corrected you have two recovery options:

- Resume the suspended re-key session. All DEK cluster or HA cluster members must be online and reachable for this command to succeed. If successful, this command resumes the re-key sessions from the point where it was interrupted.
 1. Enter the **cryptocfg --resume_rekey** command, followed by the CryptoTarget container name, the LUN number and the initiator PWWN.

```
FabricAdmin:switch>cryptocfg --resume_rekey my_disk_tgt 0x0 \
10:00:00:05:1e:53:37:99
Operation Succeeded
```

2. Check the status of the resumed re-key session.

6 MPIO and internal LUN states

```
FabricAdmin:switch> cryptocfg --show -rekey -all
```

- Read all data off the LUN and write it to another LUN. In this case, you can cancel the re-key session by removing the LUN from its container and force committing the transaction. Refer to the section [“Removing a LUN from a CryptoTarget container”](#) on page 111 for instructions on how to remove a LUN by force.

State and Status Information

In this appendix

- [Encryption engine security processor \(SP\) states](#) 189
- [Security processor KEK status](#) 190
- [Encrypted LUN states](#) 190

Encryption engine security processor (SP) states

Table 17 lists the encryption engine security processor (SP) states.

TABLE 17 Encryption engine security processor (SP) states

Encryption engine security processor (SP) state	Description
Not available	SLOT_XXX_SCN has not been received (debug message).
Not Brocade Encryption Switch or DCX	Debug message.
Not Ready	BLADE_RDY_SCN has not been received (debug message).
Fail to connect to blade	Encryption engine is not connected to the blade processor.
Starting	BLADE_RDY_SCN received and initializing.
SPC Connecting to SP	Establishing connection to CP.
SPC Connected to SP	Connection to CP established.
SP Initialized	SP initialization completed.
Waiting for initnode	SP is awaiting initialization. Run initnode .
Disabled	SP is disabled.
Encryption engine Faulty	Encryption engine is faulty (BP fault or SP fault). Issue reboot .
Waiting for initEE	SP is awaiting initialization. Run initEE .
Waiting for regEE	SP has its own certificates but is awaiting FIPS certificates. Run regEE .
Waiting for enableEE	Awaiting the explicit enabling of encryption engine. Run enableEE .
Operational; Not Online	Encryption engine is operational but offline.
Operational; Need Valid KEK	No current master key or primary or secondary link key. Check the KEK status for more details.
Operational; Need Encryption Group	Encryption engine is operational, but EG is not configured or EG information is not available. Check EG status.
Online	Encryption engine is online.
Zeroized	Encryption engine is zeroized
INVALID	Encryption engine is invalid

Security processor KEK status

Table 18 lists security processor KEK status information.

TABLE 18 Security processor KEK status

KEK type	KEK status ¹	Description
Primary KEK (current MK or primary KV link key)	None	Primary KEK is not configured.
	Mismatch	Primary KEK mismatch between the CP and the SP.
	Match/Valid	Primary KEK at CP matches the one in the SP and is valid.
Secondary KEK (alternate MK or secondary KV link key)	None	Secondary KEK is not configured.
	Mismatch	Secondary KEK mismatch between the CP and the SP.
	Match/Valid	Secondary KEK at CP matches the one in the SP and is valid.
Group KEK	None	Group KEK is not configured.
	Mismatch	Group KEK mismatch between the CP and the SP.
	Match/Valid	Group KEK at the CP matches the one in the SP and is valid.

1. Only valid in the “encryption engine awaiting encryption group” state and the “encryption engine online” state.

Encrypted LUN states

Table 19 lists encrypted LUN states. Table 20 lists LUN states that are specific to tape LUNs.

TABLE 19 Encrypted LUN states

LUN state	String displayed
UNKNOWN	Unknown
LUN_STATE_UNAVAILABLE	LUN state unavailable.
LUN_STATE_INIT	Initialize
LUN_DISC_START	LUN discovery in progress.
LUN_DISC_COMPLETE	LUN discovery complete.
LUN_SETUP_START	LUN setup
LUN_CLEAR_TEXT	cleartext encryption enabled.
LUN_ENCRYPT	Encryption enabled.
LUN_READONLY_1	Read only (found native metadata while LUN is in DF mode).
LUN_READONLY_2	Read only (found DF metadata while LUN is in native mode).
LUN_READONLY_3	Read only (metadata key is in read-only state).
LUN_WR_META_IN_PROG	Write metadata is in progress.

TABLE 19 Encrypted LUN states (Continued)

LUN_1ST_TIME_REKEY_IN_PROG	First time re-key is in progress.
LUN_KEY_EXPR_REKEY_IN_PROG	Key expired re-key is in progress.
LUN_MANUAL_REKEY_IN_PROG	Manual re-key is in progress.
LUN_DECRYPT_IN_PROG	Data decryption is in progress.
LUN_WR_META_PENDING	Write metadata is pending.
LUN_1ST_TIME_REKEY_PENDING	First time re-key is pending.
LUN_KEY_EXPR_REKEY_PENDING	Key expired re-key is pending.
LUN_MANUAL_REKEY_PENDING	Manual re-key is pending.
LUN_DECRYPT_PENDING	Data decryption is pending.
LUN_LOGIN_REQ	Login in progress.
LUN_LOGIN_BUSY	Login busy.
LUN_LOGIN_TIMEOUT	Login timeout.
LUN_ACCESS_DENIED	Login failure.
LUN_TGT_OFFLINE	Target offline.
LUN_ACCESS_CHK	Not ready (Read/Write access verification in progress).
LUN_DISCOVERY_FAILURE	LUN discovery failure.
LUN_DIS_DEK_GET_API_ERR	Disabled (Get key record API returns error).
LUN_DIS_DEK_GET_CB_ERR	Disabled (Key retrieval from vault failed).
LUN_DIS_DEK_INJECT_API_ERR	Disabled (Inject key API returns error).
LUN_DIS_DEK_INJECT_CB_ERR	Disabled (key injection failure).
LUN_DIS_BAD_KEY_STATE_1	Disabled (New key is in re-key state but encrypt exist data is off).
LUN_DIS_META_KEY_NOT_FOUND	Disabled (unable to retrieve key by key ID found from metadata).
LUN_DIS_META_KEY_MISMATCH_1	Disabled (Meta key is in re-key state but it is not the newest key).
LUN_DIS_META_KEY_MISMATCH_2	Disabled (Meta key does not match with one key found by LUN SN).
LUN_DIS_META_KEY_MISMATCH_3	Disabled (Meta key does not match with any key found by LUN SN).
LUN_DIS_BAD_META_KEY_STATE_1	Disabled (Meta key is old key and in rd/wr but new key is not in re-key).
LUN_DIS_NO_CFG_KEY_ID	Disabled (Data state is encrypted but no key ID provided and metadata does not exist).
LUN_DIS_CREATE_KEY_API_ERR	Disabled (Create new key API returns error).
LUN_DIS_CREATE_KEY_CB_ERROR	Disabled (Create new key failure).
LUN_DIS_ADD_KEY_API_ERR	Disabled (Add new key API returns error).
LUN_DIS_ADD_KEY_CB_ERR	Disabled (Add new key failure).
LUN_DIS_REKEY_ACK_ERR	Disabled (Re-key back with failure).
LUN_DIS_REKEY_DONE_ERR	Disabled (Re-key done with failure).
LUN_DIS_WR_META_ACK_ERR	Disabled (Write metadata back with failure).

TABLE 19 Encrypted LUN states (Continued)

LUN_DIS_WR_META_DONE_ERR	Disabled (Write metadata done with failure).
LUN_DIS_LUN_REMOVED	Disabled (LUN re-discovery detects LUN is removed).
LUN_DIS_LSN_MISMATCH	Disabled (LUN re-discovery detects new device ID).
LUN_DIS_DUP_LSN	Disabled (Duplicate LUN SN found).
LUN_DIS_DISCOVERY_FAIL	Disabled (LUN discovery failure).
LUN_DIS_NO_LICENSE	Disabled (Third party license is required).
LUN_DIS_WRONG_DEV_TYPE	Disabled (Wrong device type found).
LUN_DIS_NOT_SUPPORTED	Disabled (LUN not connected or supported).
LUN_DIS_CFG_KEY_NOT_FOUND	Disabled (Unable to retrieve key by key ID specified from configuration).
LUN_DIS_META_FOUND	Disabled (Data state is cleartext but metadata exists on the LUN).
LUN_DIS_BAD_KEY_STATE_2	Disabled (Data state is encrypted but there is one key which is in re-key state).
LUN_DIS_BAD_KEY_STATE_3	Disabled (Key is in invalid re-key state for encrypted data).
LUN_DIS_BAD_KEY_STATE_4	Disabled (Key is in invalid re-key state while there is one key).
LUN_DIS_BAD_KEY_STATE_5	Disabled (Key is in unknown re-key state).
LUN_DIS_NO_LICENSE_2	Disabled (Found DF metadata while LUN is in native mode and third party license is disabled).
LUN_DIS_LUN_NOT_FOUND	Disabled (LUN not found).
LUN_DIS_GET_DEV_TYPE	Disabled (Inquiry fails).
LUN_DIS_GET_DEV_ID	Disabled (Inquiry device ID page fails).
LUN_DIS_META_FOUND_2	Disabled (Found metadata while LUN is cleartext).
LUN_STATE_UNKNOWN	State of the LUN is unknown.

TABLE 20 Tape LUN states

Internal Names	Console String	Explanation
LUN_DIS_LUN_NOT_FOUND	Disabled (LUN not found)	No logical unit structure in tape module. This is an internal software error. If it occurs, contact Brocade support.
LUN_TGT_OFFLINE	Target Offline	Target port is not currently in the fabric. Check connections and L2 port state.
LUN_DIS_NOT_SUPPORTED	Disabled (LUN not connected or supported)	The target port is active, but this particular Logical Unit is not supported by that target. This indicates a user configuration error.
LUN_DIS_WRONG_DEV_TYPE	Disabled (Wrong device type found)	The logical unit on target port is active, but it is neither a tape or a medium changer. This indicates a user configuration error.
LUN_MEDIUM_CHANGER_ACTIVE	Tape medium changer active	The logical unit is a medium changer, fully ready to handle tapes.
LUN_VALIDATION_PENDING	Tape validation pending	<p>The logical unit is either a tape drive or an attached medium changer, where changer and tape are on same LUN. Since the last LOAD, REWIND, or UNIT ATTENTION, no host has attempted to read or write to a tape in this logical unit. There is no way of knowing if a tape is still present, or the encryption state of the tape.</p> <p>A host can issue a READ or WRITE to the logical unit. At that point, it can be determined whether or not a tape is present or needs to be mounted, and whether or not data is ciphertext (encrypted) or cleartext.</p>
LUN_VALIDATION_IN_PROGRESS	Tape validation in progress	The tape module has received the READ or WRITE command that triggers the validation of tape encryption mode, and is in the process of figuring out if the mounted tape medium is encrypted or not. This state should only appear briefly.
LUN_CLEAR_TEXT	Clear text	The tape medium is present, and is in clear text. The encryption switch or blade has full read/write access, because its current tape policy for the medium is also clear text.

TABLE 20 Tape LUN states

LUN_ENCRYPT	Encryption enabled	The tape medium is present, and is in ciphertext (encrypted). The encryption switch or blade has full read/write access, because its current tape policy for the medium is also encrypted. See the Encryption Format field to find out if tape is encrypted in native mode or DataFort-compatible mode.
LUN_DIS_NO_LICENSE	Disabled (Third party license is required)	The tape medium or its current tape policy is DataFort-compatible mode, but The encryption switch or blade does not have the appropriate license to enable this feature. The tape medium is neither readable nor writable.
LUN_CFG_MISMATCH_CLEARTEXT	Read only (Cleartext tape, policy mismatch)	The tape medium is clear text, but current tape policy is not. Mixed modes are not allowed, so the medium is only readable. Attempts to write result in a RASLOG and ABORTED COMMAND returned to host.
LUN_CFG_MISMATCH_COMPATIBLE	Read only (DF_compatible tape, policy mismatch)	The tape medium is encrypted and DataFort-compatible, but the current tape policy is not. Mixed modes are not allowed, so the medium is only readable. Attempts to write result in a RASLOG and ABORTED COMMAND returned to host.
LUN_CFG_MISMATCH_NATIVE	Read only (Native encrypted tape, policy mismatch))	The tape medium is encrypted and native-mode, but the current tape policy is not. Mixed modes are not allowed, so the medium is only readable. Attempts to write result in a RASLOG and ABORTED COMMAND returned to host.

LUN Policies

In this appendix

The following topics are covered in this appendix:

- [DF-compatibility support for disk LUNs](#) 195
- [DF-compatibility support for tape LUNs](#) 199

DF-compatibility support for disk LUNs

[Table 21](#) and [Table 22](#) may be used as a reference for establishing disk LUN policies in support of DataFort firmware versions.

TABLE 21 DataFort compatibility support matrix for disk LUNs

DataFort firmware versions	Brocade handling for DF disk LUNs - Read	Brocade handling for DF-compatible encryption - Write
Version 1.x	The encryption switch will support reading and decrypting the disk LUNs encrypted by this version of DataFort. The DF-compatible license is required.	The encryption switch will not support writing disk LUNs in this version format. Only the DataFort firmware version 3.x compatible encryption (metadata and algorithm) is supported for writing and encrypting the disk LUNs. This requires DF-compatible encryption mode to be set on the LUN and a DF-compatible license.
Version 2.x (No Metadata on the LUN)	The encryption switch will support reading and decrypting the disk LUNs encrypted by this version of DataFort. The DF-compatible license is required.	The encryption switch will not support writing disk LUNs in this version format. Only the DataFort firmware version 3.x compatible encryption (metadata and algorithm) is supported for writing and encrypting the disk LUNs. This requires DF-compatible encryption mode to be set on the LUN and a DF-compatible license.
Version 3.x	The encryption switch will support reading and decrypting disk LUNs encrypted by this version of DataFort. The DF-compatible license is required.	The encryption switch will support writing and encrypting the disk LUNs in this version format when DF-compatible encryption mode is set and DF-compatible License is present. Note: Brocade also supports creating new DataFort version 3.x LUNs.

B DF-compatibility support for disk LUNs

TABLE 22 Support matrix for disk LUNs for various configuration and modify options

LUN encryption format	LUN state	LUN policy	Encrypt existing data	Key ID	Metadata on LUN	Results
Native (Brocade)	Encrypted	Encrypt	NA when LUN State = encrypt	NA	Yes	No error. If the LUN was previously DF-encrypted, the LUN is set to Read Only until you either remove the LUN and add it back with the native Brocade encryption format, or issue the runtime CLI command to force the change.
Native (Brocade)	Encrypted	Encrypt	NA when LUN State = encrypt	None	No	The data encryption key is retrieved from the key vault based on the LUN serial number, and used for further encryption and decryption. An attempt is made to write the metadata. If the key cannot be retrieved for this LUN based on the LUN serial number, then the LUN is disabled for encryption. You need to either modify the LUN state to cleartext or provide the key ID in the LUN setup. You can also use the runtime cryptocfg --enable -LUN command to force the change, in which case a new key is generated and an attempt is made to write metadata.
Native (Brocade)	Encrypted	Encrypt	NA when LUN State = encrypt	Provided	No	No error.
Native (Brocade)	Encrypted	Cleartext	NA when LUN State = encrypt	NA	Yes	The LUN is disabled for encryption. Metadata is present on the LUN and the LUN is in encrypted state. You need to either modify the LUN policy to encrypt, or use the runtime cryptocfg --enable -LUN command to force the change from encrypt to cleartext.
Native (Brocade)	Encrypted	Cleartext	NA when LUN State = encrypt	None	No	No error.
Native (Brocade)	Encrypted	Cleartext	NA when LUN State = encrypt	Provided	No	The KeyID is not valid when this combination is used in cryptocfg --modify -LUN . When issuing cryptocfg --add -LUN , this is an invalid combination
Native (Brocade)	Cleartext	Encrypt	Yes	NA	Yes	The LUN is disabled for encryption. Metadata is present on the LUN and the LUN is in encrypted state. You need to either modify the LUN state to “encrypted” or use the runtime cryptocfg --enable -LUN command to force the change from the current state of the LUN to encrypt.
Native (Brocade)	Cleartext	Encrypt	Yes	None	No	No error. First time encryption started to convert the LUN from cleartext to encrypt.
Native (Brocade)	Cleartext	Encrypt	Yes	Provided	No	No Error. Key ID is ignored.

TABLE 22 Support matrix for disk LUNs for various configuration and modify options (Continued)

LUN encryption format	LUN state	LUN policy	Encrypt existing data	Key ID	Metadata on LUN	Results
Native (Brocade)	Cleartext	Cleartext	NA in case of cleartext policy	NA	Yes	The LUN is disabled for encryption. Metadata is present on the LUN and the LUN is in encrypted state. You need to either modify the LUN state to "encrypted" and change the policy to "encrypt" or use the runtime cryptocfg --enable -LUN command to force the change from the current state of the LUN to cleartext.
Native (Brocade)	Cleartext	Cleartext	NA in case of cleartext policy	None	No	No error.
Native (Brocade)	Cleartext	Cleartext	NA in case of cleartext policy	Provided	No	Error. The key ID input is not applicable to cleartext. Error is returned from the CLI.
DF compatible	Encrypted	Encrypt	NA when LUN State = encrypt	NA	Yes	No error. If the LUN was previously Brocade encrypted, the LUN is set to Read Only until you either modify the encryption format or use the runtime cryptocfg --enable -LUN command to force the change.
DF compatible	Encrypted	Encrypt	NA when LUN State = encrypt	None	No	The LUN is disabled for encryption. The key ID is missing from the user input. You need to either modify the LUN state to cleartext or provide the key ID in the LUN setup.
DF compatible	Encrypted	Encrypt	NA when LUN State = encrypt	Provided	No	No error. An attempt is made to write the metadata.
DF compatible	Encrypted	Cleartext	NA when LUN State = encrypt	NA	Yes	Adding a LUN with "cleartext" policy is invalid if the encryption format is DF-Compatible, The system returns the following error: "Bad combination of LUN options specified". The same is true for changing a LUN in DF-compatible format from "encrypt" to "cleartext" policy.
DF compatible	Encrypted	Cleartext	NA when LUN State = encrypt	None	No	Adding a LUN with "cleartext" policy is invalid if the encryption format is DF-Compatible, The system returns the following error: "Bad combination of LUN options specified". The same is true for changing a LUN in DF-compatible format from "encrypt" to "cleartext" policy.
DF compatible	Encrypted	Cleartext	NA when LUN State = encrypt	Provided	No	Adding a LUN with "cleartext" policy is invalid if the encryption format is DF-Compatible, The system returns the following error: "Bad combination of LUN options specified". The same is true for changing a LUN in DF-compatible format from "encrypt" to "cleartext" policy.

B DF-compatibility support for disk LUNs

TABLE 22 Support matrix for disk LUNs for various configuration and modify options (Continued)

LUN encryption format	LUN state	LUN policy	Encrypt existing data	Key ID	Metadata on LUN	Results
DF compatible	Cleartext	Encrypt	Yes	NA	Yes	The LUN is disabled for encryption. Metadata is present on the LUN and the LUN is in encrypted state. You need to either modify the LUN state to encrypted or use the runtime cryptocfg --enable -LUN command to force the change from the current state of the LUN to encrypt.
DF compatible	Cleartext	Encrypt	Yes	None	No	No error. First time encryption started to convert the LUN from cleartext to encrypt.
DF compatible	Cleartext	Encrypt	Yes	Provided	No	No error. The key ID is ignored.
DF compatible	Cleartext	Cleartext	NA in case of cleartext policy	NA	Yes	Not a valid combination. Cleartext support in DF-compatibility mode is rejected from the CLI.
DF compatible	Cleartext	Cleartext	NA in case of cleartext policy	None	No	Not a valid combination. cleartext support in DF-compatibility mode is rejected from the CLI.
DF compatible	Cleartext	Cleartext	NA in case of cleartext policy	Provided	No	Not a valid combination. cleartext support in DF-compatibility mode is rejected from the CLI.
Native (Brocade)	Cleartext	Encrypt	No	NA	Yes	The LUN is disabled for encryption. Metadata is present on the LUN and the LUN is in encrypted state. You need to either modify the LUN state to “encrypted” or use the runtime cryptocfg --enable -LUN command to force the change from the current state of the LUN to encrypt.
Native (Brocade)	Cleartext	Encrypt	No	NA	No	No error. Host I/O is encrypted with the new key and written to the LUN.
DF compatible	Cleartext	Encrypt	No	NA	Yes	The LUN is disabled for encryption. Metadata is present on the LUN and the LUN is in encrypted state. Modify the LUN state to “encrypted” or use the runtime cryptocfg --enable -LUN command to force the change from the current state of the LUN to encrypt.
DF compatible	Cleartext	Encrypt	No	NA	No	No error. Host I/O is encrypted with the new key and written to the LUN.

DF-compatibility support for tape LUNs

Table 23 and Table 24 may be used as a reference for establishing tape LUN policies in support of DataFort firmware versions.

NOTE

On tapes written in DataFort format, the encryption switch or blade cannot read and decrypt files with a block size of one MB or greater.

TABLE 23 Compatibility matrix for Brocade and DataFort encryption modes for tape LUNs

DataFort firmware versions	Brocade handling for DataFort written tapes - Read	Brocade handling for DataFort-compatible encryption - Write
DF SAN version 1.x	1.x tape support in DF-compatible mode is not supported in Fabric OS v6.1.1_enc.	
DF SAN version 2.x/3.x	The encryption switch supports reading and decrypting tapes of this format when a DF-compatible license is present.	The encryption switch supports writing tapes in this version format when DF-compatible encryption mode is set and a DF-compatible license is present.

TABLE 24 Compatibility support matrix for tape pools

Tape pool encryption format	Tape pool policy	Metadata present	Results
Native (Brocade)	Encrypt	Brocade metadata	No error. Both read and writes are allowed in Brocade format. The key from the metadata is used for read. A new key is generated for write if the key from the metadata has expired.
Native (Brocade)	Encrypt	DF metadata	Reads are allowed in DF-compatible format using the key from the metadata. Writes are rejected if the tape is not positioned at the beginning of the tape. Writes are allowed in Brocade format only.
Native (Brocade)	Encrypt	No (new tape)	No error. A new key is generated and both read and write are allowed in Brocade format.
Native (Brocade)	Cleartext	Brocade metadata	Reads are allowed in Brocade format using the key from the metadata. Writes are rejected if the tape is not positioned at the beginning of the tape. Writes are allowed in cleartext format (no key generated) only when the tape is positioned at the beginning of the tape.
Native (Brocade)	Cleartext	DF metadata	Reads are allowed in DF-compatible format using the key from the metadata. Writes are rejected if the tape is not positioned at the beginning of the tape. Writes are allowed in cleartext format (no key generated) only when the tape is positioned at the beginning of the tape.
Native (Brocade)	Cleartext	No (new tape)	No error. No key is generated, and both read and writes are allowed in cleartext format.
DF-compatible	Encrypt	Brocade metadata	Reads are allowed in Brocade format using the key from the metadata. Writes are rejected if the tape is not positioned at the beginning of the tape. Writes are allowed in DF-compatible format only when the tape is positioned at the beginning of the tape.
DF-compatible	Encrypt	DF metadata	No error. Both read and writes are allowed in DF-compatible format. The key from the metadata is used for read. A new key is used for write if the key from the metadata has expired.

B DF-compatibility support for tape LUNs

TABLE 24 Compatibility support matrix for tape pools (Continued)

Tape pool encryption format	Tape pool policy	Metadata present	Results
DF-compatible	Encrypt	No (new tape)	No error. A new key is generated and both read and write are allowed in DF-compatible format.
DF-compatible	Cleartext	Brocade metadata	Reads are allowed in Brocade format using the key from the metadata. Writes are rejected if the tape is not positioned at the beginning of the tape. Writes are allowed in cleartext format (no key generated) only when the tape is positioned at the beginning of the tape.
DF-compatible	Cleartext	DF metadata	Reads are allowed in DF-compatible format using the key from the metadata. Writes are rejected if the tape is not positioned at the beginning of the tape. Writes are allowed in cleartext format (no key generated) only when the tape is positioned at the beginning of the tape.
DF-compatible	Cleartext	No (new tape)	No error. No key is generated, and both read and writes are allowed in cleartext format.

NS-Based Transparent Frame Redirection

Table 25 provides the NS-based transparent frame redirection interoperability matrix.

TABLE 25 NS-based transparent frame redirection interoperability matrix¹

Frame redirection support	FOS version	Host and target edge switches/directors			
		FOS only	FOS and EOSc and EOSn interop mode 2 "native"	FOS and EOSc and EOSn interop mode 3 "open"	EOSc and EOSn only
Layer 2 SAN	FOS 6.2.0	FOS 5.3.1x for legacy Bloom-based switches and directors. FOS 6.1.1 for Condor and Condor-2 based switches and directors.	FOS 6.1.1 and EOSc 9.8 or later, and EOSn 9.8 or later.	FOS 6.2.0 and EOSc 9.9 and EOSn 9.9.	Not applicable as the encryption switch with EOSc and EOSn fabric becomes interop mode 2 or 3.
Layer 3 MetaSAN (FCR)	FOS 6.2.0	FOS 5.3.1x for legacy Bloom-based switches and directors. FOS 6.1.1 for Condor and Condor-2 based switches and directors.	FOS 6.2.0 and EOSc 9.9 and EOSn 9.9. Also requires a FOS switch with firmware version 6.2.0 or later in the fabric to manually create redirection zones.	FOS 6.2.0 and EOSc 9.9 and EOSn 9.9. Also requires a FOS switch with firmware version 6.2.0 or later in the fabric to manually create redirection zones.	EOSc 9.8 or later, and EOSn 9.8 or later. Also requires a FOS switch with firmware version 6.1.1 or later in the fabric to manually create redirection zones.

1. Only the M6140, M4700F, McDATA 4400, and Brocade Intrepid 10000 are supported for frame redirection.

NOTE

When an EOSc switch is powered down and powered up again, redirection zone information is erased. No devices are allowed to log in at this stage. To enable all devices to log in, issue a **cfgsave** command from the FOS switch where the redirection zones were originally created, or issue an **rd zone delete** command from EOSc switch.

C NS-Based Transparent Frame Redirection

Supported Key Management Systems

In this appendix

- [Key management systems](#) 203
- [The NetApp Lifetime Key Manager](#)..... 204
- [The RSA Key Manager](#)..... 212
- [The HP Secure Key Manager](#) 218
- [Thales Encryption Manager for Storage](#)..... 232

Key management systems

Data is encrypted and decrypted using the same Data encryption key (DEK), so a DEK must be preserved at least long enough to decrypt the ciphertext that was created using that DEK. The length of time data is stored before it is retrieved can vary greatly. Some data may be stored for months, years or decades before it is accessed. To be sure encrypted data remains accessible DEKs also need to be stored for months, years or decades. This requires the use of a key management system.

Key management systems are available from several vendors to provide life cycle management for all DEKs created by the encryption engine. The following key management systems currently support Brocade encryption switches and blades:

- NetApp Lifetime Key Manager (LKM).
- RSA Key Manager (RKM).
- Hewlett Packard Secure Key Manager (HP SKM).
- Thales Encryption Manager for Storage (TEMS), also referred to as the nCipher Key Authority (NCKA) within operational descriptions in this document.

The NetApp Lifetime Key Manager

The NetApp Lifetime Key Manager (LKM) resides on an FIPS 140-2 Level 3-compliant network appliance. The encryption engine and LKM appliance communicate over a trusted link. A trusted link is a secure connection established between the Encryption switch or blade and the NetApp LKM appliance, using a shared secret called a link key. One link key per encryption switch is established with each LKM appliance. On a Brocade DCX or DCX-4S or with one or two FS8-18 encryption blades, only one link key is established with each LKM appliance, and the link key is shared between the blades.

DEKs are encrypted by the encryption engine, using its link key, and passed to LKM over a secure connection. LKM decrypts the DEKs and encrypts them on the LKM appliance. When the encryption engine needs a DEK from the LKM key vault, it passes a request that includes a key ID and other parameters needed by LKM to locate the correct key. LKM locates the DEK, decrypts it, and then encrypts it using its key for transfer to the encryption engine.

Setting up an LKM key vault consists of the following steps:

- Authenticating the NetApp LKM appliance with the group leader by registering certificates containing the public key and IP address with the group leader. The group leader automatically distributes the certificate and the IP address of the NetApp LKM appliance to all group members.
- Authenticating the encryption group leader and each encryption group member with the NetApp LKM appliance. For each node in the encryption group, the IP address and the certificate containing the public key are registered with the NetApp LKM appliance. The registered certificate is a special purpose KAC Certificate that contains license information related to the LKM.
- Establishing a trusted link between the NetApp LKM appliance and each member node. As part of the trusted link establishment, a shared secret called a link key is created on each of the two entities. The link key is subsequently used for encrypting the DEKs for archival to the NetApp LKM appliance or for decrypting the encrypted DEKs for retrieval from the NetApp LKM appliance.

The NetApp DataFort Management Console

The NetApp DataFort Management Console (DMC) must be installed on your PC or workstation to complete certain procedures described in this appendix. Refer to the appropriate DMC product documentation for DMC installation instructions. After you install DMC, do the following.

1. Launch the DMC.
2. Click the **Appliance** tab on the top panel.
3. Add the NetApp LKM appliance IP address or hostname.
4. Right-click the added IP address and log into the NetApp LKM key vault.

Obtaining and importing the LKM certificate

Certificates must be exchanged between LKM and the encryption switch to enable mutual authentication. You must obtain a certificate from LKM, and import it into the encryption group leader. The encryption group leader exports the certificate to other encryption group members.

To obtain and import an LKM certificate, do the following.

1. Open an SSH connection to the NetApp LKM appliance and log in.

```
host$ssh admin@10.33.54.231
admin@10.33.54.231's password:

Copyright (c) 2001-2009 NetApp, Inc.
All rights reserved
+-----+
| NetApp Appliance Management CLI |
|           Authorized use only!   |
+-----+
Cannot read termcapdatabase;
using dumb terminal settings.
Checking system tamper status:
No physical intrusion detected.
```

2. Add the group leader to the LKM key sharing group. Enter **lkmserver add --type third-party --key-sharing-group "/"** followed by the group leader IP address.

```
lkm-1>lkmserver add --type third-party --key-sharing-group \
"/" 10.32.244.71
NOTICE: LKM Server third-party 10.32.244.71 added.
Cleartext connections not allowed.
```

3. On the NetApp LKM appliance terminal, enter **sys cert getcert-v2** to display the LKM certificate content.

```
lkm-1> sys cert getcert-v2
-----BEGIN CERTIFICATE-----
[content removed]
-----END CERTIFICATE-----
```

4. Copy and paste the LKM certificate content from the NetApp LKM appliance terminal into an editor buffer. Save the file as **lkmcert.pem** on the SCP-capable host. Save the entire certificate, including the lines **-----BEGIN CERTIFICATE-----** and **-----END CERTIFICATE-----**.

5. On the group leader, import the previously saved LKM certificate from the SCP-capable host:
 - If you are using DCFM, the path to the file must be specified on the **Select Key Vault** dialog box. If the proper path is entered, the file is imported.
 - If you are using the CLI, use the **cryptocfg – import** command with the **-scp** option. The following example imports a certificate file named **lkmcert.pem**.

```
SecurityAdmin:switch>cryptocfg --import -scp lkmcert.pem 192.168.38.245 \
mylogin /tmp/certs/lkmcert.pem

Password:
Operation succeeded.
```

Registering the certificates

The switch's KAC certificate must be registered on the LKM appliance, and the LKM certificate must be registered on the switch.

1. From the external host, register the KAC certificate you exported from the group leader with the NetApp LKM appliance.

```
host$echo lkmserver certificate set 10.32.244.71 \  
'cat kac_lkm_cert.pem' | ssh -l admin 10.33.54.231  
Pseudo-terminal will not be allocated because stdin is not a terminal.  
admin@10.33.54.231's password:  
Checking system tamper status:  
No physical intrusion detected.  
NOTICE: LKM Peer '10.32.244.71' certificate is set
```

2. On the group leader, register the NetApp LKM appliance as the primary key vault LKM1.

```
SecurityAdmin:switch>cryptocfg --reg -keyvault LKM1 lkmcert.pem \  
10.33.54.231 primary  
lkm-1  
Register key vault status: Operation Succeeded.
```

3. Display the registered key vault on the group leader. The LKM key vault is shown as connected.

```
SecurityAdmin:switch>cryptocfg --show -groupcfg  
Encryption Group Name:      brocade  
Failback mode:             Manual  
Heartbeat misses:          3  
Heartbeat timeout:         2  
Key Vault Type:            LKM  
Primary Key Vault:  
IP address:                 10.33.54.231  
Certificate ID:             lkm-1  
Certificate label:          LKM1  
State:                      Connected  
Type: LKM  
Secondary Key Vault not configured  
NODE LIST  
Total Number of defined nodes: 2  
Group Leader Node Name:     10:00:00:05:1e:41:7e  
Encryption Group state:     CLUSTER_STATE_CONVERGED  
Node Name                   IP address      Role  
10:00:00:05:1e:41:9a:7e     10.32.244.71   GroupLeader(current node)  
10:00:00:05:1e:39:14:00     10.32.244.60   MemberNode
```

4. Display the registered key vault on the member node. The LKM key vault is shown as not responding because certificates have not been exchanged.

```
SecurityAdmin:encl_switch>cryptocfg --show -groupcfg  
Encryption Group Name:      brocade  
Failback mode:             Manual  
Heartbeat misses:          3  
Heartbeat timeout:         2  
Key Vault Type:            LKM  
Primary Key Vault:  
IP address:                 10.33.54.231  
Certificate ID:             lkm-1  
Certificate label:          LKM1  
State:                      Not responding  
Type: LKM  
Secondary Key Vault not configured
```



```

NODE LIST
Total Number of defined nodes: 2
Group Leader Node Name:      10:00:00:05:1e:41:7e
Encryption Group state:     CLUSTER_STATE_CONVERGED
Node Name                    IP address      Role
10:00:00:05:1e:41:9a:7e     10.32.244.71   GroupLeader
10:00:00:05:1e:39:14:00     10.32.244.60   MemberNode (current node)

```

5. Exchange certificates between the LKM key vault and the member node, starting with exporting the KAC certificate from the member node to an SCP-capable external host.

```

SecurityAdmin:encl_switch>cryptocfg --export -scp -KACcert \
192.168.38.245 mylogin encl_kac_lkm_cert.pem
Password:
Operation succeeded.

```

6. Open an SSH connection to the NetApp LKM appliance and add the member node IP address.

```

lkm-1> lkmserver add --type third-party --key-sharing-group "/" \
10.32.244.60
NOTICE: LKM Server third-party 10.32.244.60 added.
Cleartext connections not allowed.

```

7. On the external host, register the KAC LKM certificate you exported from the member node with the NetApp LKM appliance.

```

host$echo lkmserver certificate set 10.32.244.60
'cat encl_kac_lkm_cert.pem' | ssh-l admin 10.33.54.231
Pseudo-terminal will not be allocated because stdin is not a terminal.
admin@10.33.54.231's password:
Checking system tamper status:No physical intrusion detected.
ALERT: There are pending unapproved trustees.
NOTICE: LKM Peer '10.32.244.60' certificate is set

```

8. Enter the `cryptocfg --show -groupcfg` command on the member node. If the link key has been established (refer to [“Establishing the trusted link”](#)), the display shows the LKM as connected.

```

SecurityAdmin:encl_switch>cryptocfg --show -groupcfg
Encryption Group Name:      brocade
  Failback mode:           Manual
  Heartbeat misses:        3
  Heartbeat timeout:       2
  Key Vault Type:          LKM
Primary Key Vault:
  IP address:              10.33.54.231
  Certificate ID:           lkm-1
  Certificate label:        LKM1
  State:                   Connected
  Type: LKM
Secondary Key Vault not configured
[output truncated]

```

Establishing the trusted link

You must generate the trusted link establishment package (TEP) on all nodes to obtain a trusted acceptance package (TAP) before you can establish a trusted link between each node and the NetApp LKM appliance. You must have a card reader attached to your PC or workstation to complete the procedure.

NOTE

Complete all steps required to establish a trusted link between LKM and the encryption group members for each node before proceeding to the next node.

1. Open an SSH connection to the NetApp LKM appliance and log in.

```
host$ssh admin@10.33.54.231
admin@10.33.54.231's password:

Copyright (c) 2001-2008 NetApp, Inc.
All rights reserved
+-----+
| NetApp Appliance Management CLI |
|       Authorized use only!       |
+-----+
Cannot read termcapdatabase;
using dumb terminal settings.
Checking system tamper status:
No physical intrusion detected.
```

2. To add the encryption group leader to an LKM appliance third party key sharing group, enter **lkmserver add --type third-party --key-sharing-group "/"** followed by the group leader IP address.

```
lkm-1>lkmserver add --type third-party --key-sharing-group \
"/" 10.32.244.71
NOTICE: LKM Server third-party 10.32.244.71 added.
Cleartext connections not allowed.
```

- From the external host, enter `echo lkmserver set <group leader IP address> 'cat kac_cert_lkm.pem' | ssh -l admin <LKM IP address>` to register the KAC LKM certificate you exported from the group leader with the NetApp LKM appliance.

```
host$echo lkmserver certificate set 10.32.244.71 \  
'cat kac_lkm_cert.pem' | ssh -l admin 10.33.54.231  
Pseudo-terminal will not be allocated because stdin is not a terminal.  
admin@10.33.54.231's password:  
Checking system tamper status:  
No physical intrusion detected.  
NOTICE: LKM Peer '10.32.244.71' certificate is set
```

- Select the **Link Keys** tab on the **Encryption Group Properties** dialog box.
The switch name displays in the link status table under **Switch**, with a **Link Key Status** of **Link Key requested, pending LKM approval**.
- Select the switch, and click **Establish**.
This results in a Trusted link establishment package (TEP), which is needed to establish the trusted link between the switch and the LKM appliance.
- Launch the NetApp DataFort Management Console (DMC) and click the **View Unapproved Trustees** tab.
The switch is listed as `openkey_trustee_<ip address>`, where the IP address is the switch IP address entered in step 2.
- Select the switch, and click **Approve and Create TAP**.
The **Approve TEP** dialog box displays. The TEP must be approved before a TAP can be created.
- Provide a label in the dialog box and click **Approve** to approve the TEP.
A list of recovery cards and recovery officers is displayed. TEP approval is done by a quorum of recovery officers, using assigned recovery cards. Each recovery officer must individually insert one of listed recovery cards into a card reader attached to the PC or workstation, enter the password for that card, and click **Start**. The procedure is repeated until a quorum of recovery officers has approved the TEP.
- Save the TAP to a file (location does not matter).
- Select the **Link Keys** tab on the **Encryption Group Properties** dialog box.
- Select the switch in the link key status table, and click **Accept** to retrieve the TAP from the LKM appliance.
- Repeat the above steps for the each of the remaining member nodes.

LKM key vault high availability deployment

LKM appliances can be clustered together to provide high availability capabilities. You can deploy and register one LKM with an encryption switch or blade and later deploy and register another LKM at any time, if LKMs are clustered or linked together. Please refer to the Release Notes for Fabric OS version 6.3.0, and LKM documentation to link or cluster the LKMs.

When LKM appliances are clustered, both LKMs in the cluster must be registered and configured with the link keys before starting any crypto operations. If two LKM key vaults are configured, they must be clustered. If only a single LKM key vault is configured, it may be clustered for backup purposes, but it will not be directly used by the switch.

When dual LKMs are used with the encryption switch or blade, the dual LKMs must be clustered. There is no enforcement done at the encryption switch or blade to verify whether or not the dual LKMs are clustered, but key creation operations will fail if you register non-clustered dual LKMs with the encryption switch or blade.

Regardless of whether you deploy a single LKM or clustered dual LKMs, register only the primary key vault with the encryption switch or blade. You do not need to register a secondary key vault.

Use the following command to register an LKM key vault on the encryption switch or blade.

```
cryptocfg --reg -keyvault <cert label> <certfile> <hostname/ip address> primary
```

Disk keys and tape pool keys (Brocade native mode support)

DEK creation, retrieval, and update for disk and tape pool keys in Brocade native mode are as follows:

- **DEK creation** - The DEK is archived into the primary LKM. Upon successful archive of DEK onto primary LKM, the DEK is read from secondary LKM until it is synchronized to the secondary LKM, or a timeout of 10 seconds occurs (2 seconds with 5 retries). If successful, then the DEK created can be used for encrypting disk LUNs or tape pool in Brocade native mode. If key archival of the DEK to primary LKM fails, an error is logged and the operation is retried. If the failure happens after archival of the DEK to the primary LKM, but before synchronization to the secondary, a VAULT_OFFLINE error is logged and the operation is retried. Any DEK archived to the primary in this case is not used.
- **DEK retrieval** - The DEK is retrieved from the primary LKM if the primary LKM is online and reachable. If the registered primary LKM is not online or not reachable, the DEK is retrieved from a clustered secondary LKM.
- **DEK Update** - DEK Update behavior is same as DEK Creation.

Tape LUN and DF-compatible tape pool support

- **DEK Creation** - The DEK is created and archived to the primary LKM only. Upon successful archival of the DEK to the primary LKM, the DEK can be used for encryption of a Tape LUN or DF-Compatible tape pool. The DEK is synchronized to a secondary LKM through LKM clustering. If DEK archival to the primary LKM fails, DEK archival is retried to the clustered secondary LKM. If DEK archival also fails to secondary LKM, an error is logged and the operation is retried.
- **DEK retrieval** - The DEK is retrieved from primary LKM if primary is online and reachable. If primary LKM is not online or not reachable, the DEK is retrieved from the clustered secondary LKM.
- **DEK update** - DEK update behavior is same as DEK Creation.

LKM Key Vault Deregistration

Deregistration of either Primary or Secondary LKM KV from an encryption switch or blade is allowed independently.

- **Deregistration of Primary LKM** - You can deregister the Primary LKM from an encryption switch or blade without deregistering the backup or secondary LKM for maintenance or replacement purposes. However, when the primary LKM is deregistered, key creation operations will fail until either primary LKM is reregistered or the secondary LKM is deregistered and reregistered as Primary LKM.

When the Primary LKM is replaced with a different LKM, you must first synchronize the DEKs from secondary LKM before reregistering the primary LKM.

- **Deregistration of Secondary LKM** - You can deregister the Secondary LKM independently. Future key operations will use only the Primary LKM until the secondary LKM is reregistered on the encryption switch or blade.

When the Secondary LKM is replaced with a different LKM, you must first synchronize the DEKs from Primary LKM before reregistering the secondary LKM.

The RSA Key Manager

Communication with the RSA Key Manager (RKM) is secured by wrapping DEKs in a master key. The encryption engine must generate its own master key, send DEKs to RKM encrypted in the master key, and decrypt DEKs received from RKM using the same master key. The master key may optionally be stored as a key record in the RKM key vault as a backup, but RKM does not assume responsibility for the master key. The master key must be backed up and stored, and policies and procedures for responding to theft or loss must be in place.

Obtaining and Importing the RKM certificate

Certificates must be exchanged between RKM and the encryption switch to enable mutual authentication. You must obtain a certificate from RKM, and import it into the encryption group leader. The encryption group leader exports the certificate to other encryption group members.

To obtain and import an RKM certificate, do the following.

1. Export the RKM certificate using a file transfer utility, such as FTP, and save it on an SCP-capable host.
2. On the group leader, import the previously saved RKM certificate from the SCP-capable host:
 - If you are using DCFM, the path to the file must be specified on the **Select Key Vault** dialog box. If the proper path is entered, the file is imported.
 - If you are using the CLI, use the **cryptocfg – import** command with the **-scp** option. The following example imports a certificate file named **rkmcert.pem**.

```
SecurityAdmin:switch>cryptocfg --import -scp rkmcert.pem 192.168.38.245 \
mylogin /tmp/certs/rkmcert.pem
Password:
Operation succeeded.
```

Exporting the KAC certificate signing request (CSR)

If you are using the SAN Management program, the KAC CSR is exported to a location you specify when you create a new encryption group or add a switch to an encryption group. If you are using the CLI, you can export the KAC CSR from the switch to file on a LAN-attached host, or you can attach a USB storage device to the switch and export the KAC CSR to that device.

1. Log into the switch on which the CSR was generated as Admin or SecurityAdmin.
2. Export the CSR from the switch over an SCP-protected LAN connection to a file on an external host (e.g., your workstation), or to a mounted USB device.

The following example exports a CSR to an external SCP-capable host.

```
SecurityAdmin:switch>cryptocfg --export -scp -KACcsr \
192.168.38.245 mylogin /tmp/certs/kac_rkm_cert.pem
Password:
Operation succeeded.
```

The following example exports a CSR to USB storage.

```
SecurityAdmin:switch>cryptocfg --export -usb KACcsr kac_rkm_cert.pem
Operation succeeded.
```

If you export the CSR to a USB storage device, you will need to remove the storage device from the switch, and then attach it to a computer that has access to a third party certificate authority (CA). If you are using the SAN Management application, this can be your SAN Management application workstation. The CSR must be submitted to a CA.

NOTE

The CSR is exported in Privacy Enhanced Mail (.pem) format. This is the format required in exchanges with certificate authorities.

Submitting the CSR to a certificate authority

The CSR must be submitted to a certificate authority (CA) to be signed. The certificate authority is a trusted third party entity that signs the CSR. There are several CAs available, and procedures vary, but the general steps are as follows.

1. Open an SSL connection to an X.509 server.
2. Submit the CSR for signing.
3. Request the signed certificate.

Generally, a public key, the signed KAC certificate, and a signed CA certificate are returned.

4. Store the signed certificates, preferably in the same location as the CSR.

Importing the signed KAC certificate

The signed KAC certificate must be imported into the switch or blade that generated the CSR.

If you are using the SAN Management program, do the following.

1. Select **Configure > Encryption** from the menu bar.

The **Encryption Center** dialog box displays the status of all encryption-related hardware and functions at a glance. It is the single launching point for all encryption-related configuration.

2. Select the switch or encryption engine from the **Encryption Devices** table, and select **Switch > Properties** or **Engine > Properties** from the menu bar, or right-click the switch or encryption engine and select **Properties**.

The **Encryption Properties** dialog box is displayed.

3. Click **Import**

An **Open** dialog box is displayed.

4. From **Look In**, browse to the location where you stored the signed KAC certificate after you received it from the CA.
5. To limit the number of files displayed to .pem files, select Certificate Files (*.pem) from **Files of Type**.
6. Select the file and click **Open**.
You are returned to **Encryption Properties**.
7. Click **Save**.

D The RSA Key Manager

If you are using the CLI, you can import the signed KAC certificate to the switch from a file on a LAN attached host, or you can write it to a USB storage device, attach the USB storage device to the switch or blade, and import the certificate from that device. The following describes both options.

1. Log into the switch to which you wish to import the certificate as Admin or SecurityAdmin.
2. Enter the **cryptocfg --import** command with the appropriate parameters.

The following example imports a CP certificate named “enc_switch1_cp_cert.pem” that was previously exported to the external host 192.168.38.245. Certificates are imported to a predetermined directory on the node.

```
SecurityAdmin:switch>cryptocfg --import -scp enc_switch1_cp_cert.pem \  
192.168.38.245 mylogin /tmp/certs/enc_switch1_cp_cert.pem  
Password:  
Operation succeeded.
```

The following example imports a CP certificate named “enc_switch1_cp_cert.pem” that was previously exported to USB storage.

```
SecurityAdmin:switch>cryptocfg --import -usb enc_switch1_cp_cert.pem \  
enc_switch1_cp_cert.pem  
Operation succeeded.
```

3. Register the KAC certificate.

```
SecurityAdmin:switch>cryptocfg --reg -KACcert <certificate file>
```


Uploading the KAC and CA certificates onto the RKM appliance

After an encryption group is created, you need to install the switch public key certificate (KAC certificate) and signing authority certificate (CA certificate) on the RKM appliance.

1. Start a web browser, and connect to the RKM appliance setup page. You will need the URL, and have the proper authority level, a user name, and a password.
2. Select the **Operations** tab.
3. Select **Certificate Upload**.
4. In the **SSLCAcertificateFile** field, enter the full local path of the CA certificate. Do not use the UNC naming convention format.
5. Select **Upload, Configure SSL, and Restart Webserver**.
6. After the web server restarts, enter the root password.
7. Open another web browser window, and start the RSA management user interface.
You will need the URL, and have the proper authority level, a user name, and a password.

NOTE

The Identity Group name used in the next step may not exist in a freshly installed RKM. To establish an Identity Group name, click the **Identity Group** tab, and create a name. The name **Hardware Retail Group** is used as an example in the following steps.

8. Select the **Key Classes** tab. For each of the following key classes, perform steps a. through h. to create the class. The key classes must be created only once, regardless of the number of nodes in your encryption group and regardless of the number of encryption groups that will be sharing this RKM.

kcn.1998-01.com.brocade:DEK_AES_256_XTS

kcn.1998-01.com.brocade:DEK_AES_256_CCM

kcn.1998-01.com.brocade:DEK_AES_256_GCM

kcn.1998-01.com.brocade:DEK_AES_256_ECB

- a. Click **Create**.
- b. Type the key name string into the **Name** field.
- c. Select **Hardware Retail Group** for **Identity Group**.
- d. Deselect **Activated Keys Have Duration**.
- e. Select **AES** for **Algorithm**.
- f. Select **256** for **Key Size**.
- g. Select the **Mode** for the respective key classes as follows:
 - XTS** for Key Class "kcn.1998-01.com.brocade:DEK_AES_256_XTS"
 - CBC** for Key Class "kcn.1998-01.com.brocade:DEK_AES_256_CCM"
 - CBC** for Key Class "kcn.1998-01.com.brocade:DEK_AES_256_GCM"
 - ECB** for Key Class "kcn.1998-01.com.brocade:DEK_AES_256_ECB"

- h. Click **Next**.
 - i. Repeat a. through h. for each key class.
 - j. Click **Finish**.
9. For each node, create an identity as follows.
 - a. Select the **Identities** tab.
 - b. Click **Create**.
 - c. Enter a label for the node in the **Name** field. This is a user-defined identifier.
 - d. Select the **Hardware Retail Group** in the **Identity Groups** field.
 - e. Select the **Operational User** role in the **Authorization** field.
 - f. Click **Browse** and select the imported certificate <name>_kac_cert.pem as the **Identity certificate**.
 - g. Click **Save**.
10. Register the RKM key vault on the group leader using the CA certificate for the CA that signed the RKM key vault certificate. The path to the file was entered in the **SSLCertificateFile** field. The group leader automatically shares this information with other group members.

```
SecurityAdmin:switch>cryptocfg --import -scp <CA certificate file>  
<host IP> <host username> <host path>
```

```
SecurityAdmin:switch>cryptocfg --reg -keyvault <CA certificate file>  
<RKM IP> primary
```

11. Display the group configuration, using the `cryptocfg -- show -groupcfg` command

RKM key vault high availability deployment

When dual RKM appliances are used for high availability, the RKM appliances must be clustered, and must operate in maximum availability mode, as described in the RKM appliance user documentation.

When dual RKM appliances are clustered, they are accessed using an IP load balancer. For a complete high availability deployment, the multiple IP load balancers are clustered, and the IP load balancer cluster exposes a virtual IP address called a floating IP address. The floating IP address must be registered on the encryption switch or blade using the `cryptocfg --reg -keyvault` command.

The secondary RKM appliance must not be registered, and also individual RKM appliance IP addresses must not be registered. The command to register a secondary RKM appliance is blocked, beginning with Fabric OS version 6.3.0.

DEK Creation

A newly created DEK is archived to the floating IP Address of the Clustered RKM appliances, or IP Load Balancer Cluster. The load balancer of the RKM Appliance Cluster routes the request to the primary RKM Appliance. The DEK gets archived to primary RKM Appliance, and then is synchronized to secondary RKM Appliance in the Cluster by the RKM Cluster Key Sync software. Upon successful archival of the DEK to RKM Cluster, the DEK can be used for encryption of a Disk LUN, tape LUN, or Tape Pool. If archival of the DEK to the RKM Cluster fails, an error is logged and the operation is retried.

DEK retrieval

The DEK is retrieved from the floating IP Address of the Clustered RKM appliances, or IP Load Balancer Cluster. If the DEK retrieval fails, then the DEK retrieval is retried.

DEK Update

DEK Update behavior is same as DEK Creation.

The HP Secure Key Manager

The HP StorageWorks Secure Key Manager (SKM) is a security appliance providing centralized key management operations. SKM runs on a stand-alone FIPS 140-2 level 2 compliant hardware platform that is isolated from the other applications, and runs a hardened operating system. SKM offers high availability, clustering and failover options.

After the required certificate file is loaded on the encryption switch, and the SKM IP addresses are configured on the encryption switch, the encryption switch automatically establishes a secure connection with SKM. Communication with SKM is secured by wrapping DEKs in a master key. The encryption engine must generate its own master key, send DEKs to SKM encrypted in the master key, and decrypt DEKs received from SKM using the same master key.

Setting up an HP SKM key vault consists of registering the encryption group leader and group member nodes with the HP SKM key vault by exporting their KAC certificates, creating a Brocade group on the SKM key vault, and taking steps on the HP SKM appliance that allow the certificates to be signed by a local certificate authority (CA) on the HP SKM appliance.

Obtaining a signed certificate from the HP SKM appliance software

The following steps describe how to get a signed certificate from the Hewlett Packard Secure Key Manager (HP SKM) appliance. You will need this information when you create a new encryption group with the HP SKM key vault, and you must obtain a signed certificate for each switch.

1. Select **Tools > Internet Options** on your Internet browser.

Click the **Advanced** tab, and select the **Use TLS 1.0** option.

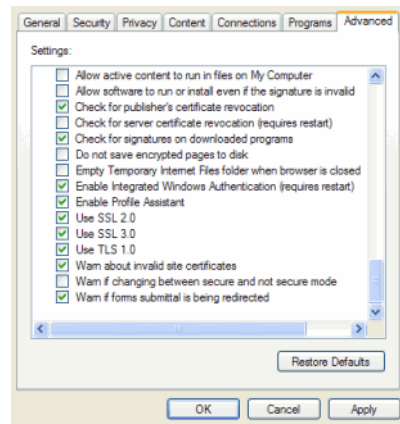


FIGURE 75 TLS 1.0 option from Internet browser

2. Log in to the HP StorageWorks Secure Key Manager appliance using a browser and https protocol:

The **HP StorageWorks Secure Key Manager Administrator Authentication** dialog box displays.

3. Enter the user name and password:

Username: admin

Password: hpskm028

The **Certificate and CA Configuration** dialog box displays.

4. Click the **Security** tab, and then click the **Sign Request** button.

The **Sign Certificate Request** dialog box displays.

5. Click the **Sign Request** button at the bottom of the screen.
6. Copy and paste the generated certificate contents from the HP SKM into a file. You will import the signed certificate into the switch in the next procedure, [“Importing a signed certificate.”](#)

Importing a signed certificate

After a signed certificate is obtained, it must be imported and registered.

1. Select a switch from the **Encryption Targets** dialog box, and click the **Properties** tab.

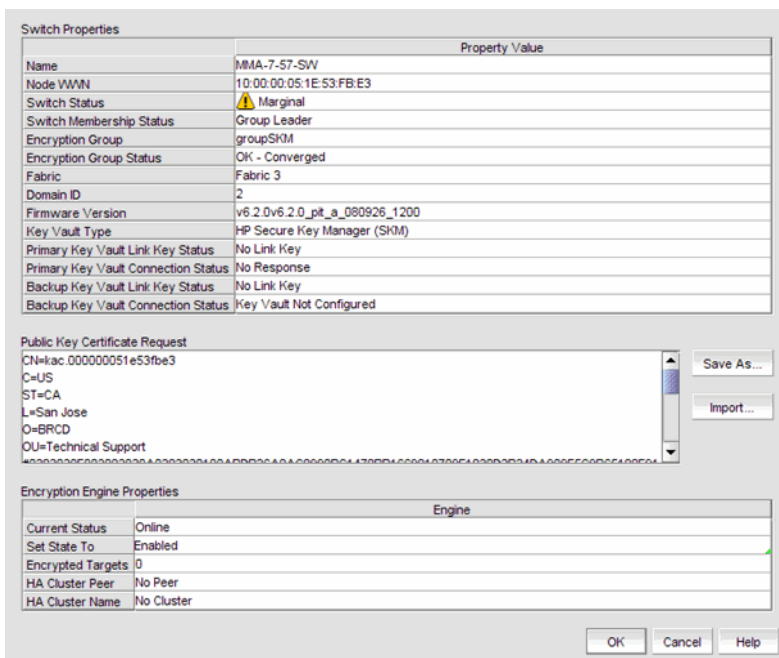


FIGURE 76 Switch Properties dialog box

2. Click the **Import** button.

The **Import Signed Certificate** dialog box displays.

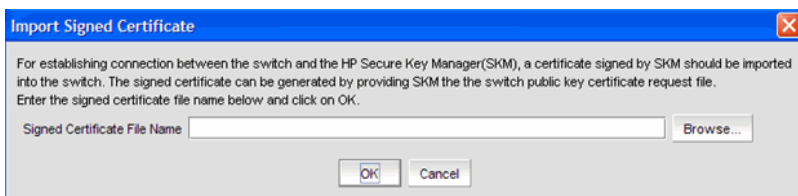


FIGURE 77 Import Signed Certificate dialog box

3. Browse to the location of the stored, signed certificate, and click **OK**.

A connection is now established between the switch and the HP Secure Key Manager (SKM).

4. Register the SKM key vault on the group leader using the CA certificate for the CA that signed the SKM key vault certificate. The group leader automatically shares this information with other group members.

```
SecurityAdmin:switch>cryptocfg --import -scp <CA certificate file>
<host IP> <host username> <host path>
```

```
SecurityAdmin:switch>cryptocfg --reg -keyvault <CA certificate file>
<RKM IP> primary
```

5. Display the group configuration, using the `cryptocfg -- show -groupcfg` command.

Exporting the KAC certificate request

A KAC certificate request must be exported for each encryption node to an SCP-capable host.

1. Log into the group leader as Admin or SecurityAdmin.
2. Set the SKM key vault type by entering the `cryptocfg --set -keyvault SKM` command with the **SKM** option. Successful execution sets the key vault type for the entire encryption group.

```
SecurityAdmin:switch>cryptocfg --set -keyvault SKM
Set key vault status: Operation Succeeded.
```

3. On each node in the encryption group, export the KAC certificate to an SCP-capable host.

```
SecurityAdmin:switch>cryptocfg --export -scp -KACcsr
192.168.38.245 mylogin /tmp/certs/kac_skm.csr
```

NOTE

Record this location so you can easily find the KAC certificate for signing in the [“Signing the KAC certificate”](#) procedure.

Configuring a Brocade group

A Brocade group is configured on SKM for all keys created by Brocade encryption switches and blades. This needs to be done only once for each key vault.

1. Launch the SKM administration console in a web browser and log in.
2. Select the **Security** tab.
3. Select **Local Users & Groups** under **Users and Groups**.
The **User & Group Configuration** page is displayed.
4. Select **Add** under **Local Users**.
5. Add a new user name under **Username**, and a password under **Password**.
6. Select the **User Administration Permission** and **Change Password Permission** check boxes.
7. Select **Save** to save this user data.
8. Select **Add** under **Local Groups**.
9. Add a new group called Brocade under **Group**.
10. Select **Save**.
11. Select the new brocade group name, and then select **Properties**.
Local **Group Properties** and a **User List** are displayed.
12. In the **User List** section, select or type the Brocade user name under **Username**.
13. Select **Save**.

The Brocade user name and password are now configured on SKM.

NOTE

Fabric OS version 6.2.0 uses brcduser1 as a standard user name when creating a Brocade group on SKM. If you downgrade from version 6.3.0 or later to version 6.2.0, the user name is overwritten to brcduser1, and the Brocade group user name must be changed to brcduser1.

Registering the Brocade user name and password in encryption groups

The Brocade group user name and password you created in “[Configuring a Brocade group](#)” must also be registered on the encryption group leader, and each node in an encryption group.

1. Starting with the encryption group leader, register the user password and user name by issuing the following command.

```
SecurityAdmin:switch>cryptocfg --reg -KAClogin primary
```

NOTE

This command is must be used only for the primary key vault.

2. When prompted, enter the user name specified in [step 5](#) of “[Configuring a Brocade group](#)”.
3. When prompted enter and confirm the password specified in [step 5](#) of “[Configuring a Brocade group](#)”.
4. Repeat the procedure for each node in the encryption group.

Keep the following rules in mind when registering the Brocade user name and password:

- The user name and password must match the user name and password specified for the Brocade group.
- The same user name and password must be configured on all nodes in an encryption group. This is not enforced or validated by the encryption group members, so care must be taken when configuring the user name and password to ensure they are the same on each node.
- Different user names and passwords can never be used within the same encryption group, but each encryption group may have its own user name and password.
- If you change the user name and password using the -KAClogin option, the keys created by the previous user become inaccessible. The Brocade group user name and password must also be changed to the same values on SKM to make the keys accessible.
- When storage is moved from one encryption group to another, and the new encryption group uses different user name and password, the Brocade group user name and password must also be changed to the same values on SKM to make the keys accessible.

Setting up the local certificate authority

The local certificate authority is set up by adding Brocade to the Local Certificate Authority List. After establishing the local certificate authority for Brocade, Brocade is then added and accepted as a trusted user of SKM.

1. Select the **Security** tab on the SKM key manager.
2. Select **Local CAs** under **Certificates and CAs**.

The **Certificate and CA Configuration** page is displayed. This page includes the **Local Certificate Authority List**, and a **Create Local Certificate Authority** dialog box.

3. Enter the following in the **Create Local Certificate Authority** dialog box:
 - Certificate Authority Name - HPSKM_CA1
 - Common Name - HPSKM_CA1
 - Organization Name - Brocade
 - Organizational Unit Name - Storage Software
 - Locality Name - SJC
 - State or Province Name - CA
 - Country Name - US
 - Email Address - support@brocade.com
 - Key Size - 2048
 - Certificate Authority Type - Select Self-Assigned Root CA. The values for CA certification Duration and Maximum User Certificate Duration should both be 3650.

NOTE

The names shown are only examples. You may use different names. Remember the **Certificate Authority Name**, or write it down. You will need later in the procedures for [“Adding the local CA to the trusted CAs list”](#), [“Adding a server certificate for the SKM appliance”](#), and [“Downloading the local CA certificate file”](#).

4. Click **Create**.

Successful completion is indicated when the new Local CA appears on the **Local Certificate Authority** List.

Adding the local CA to the trusted CAs list

You must now update the Trusted CAs list with the local CA name you created in [“Setting up the local certificate authority”](#).

1. Select the **Security** tab on the SKM key manager.
2. Select **Trusted CA Lists** under **Certificates and CAs**.

The **Trusted CA Lists** page is displayed.
3. Select **Default** under **Profile Name**.
4. Click **Properties**.

A properties dialog box is displayed.
5. Click **Edit**.

A dialog box is displayed that allows you to **Add** CAs to a **Trusted CAs** list from a list of **Available CAs**, or to **Remove** CAs from the **Trusted CAs** list and place them in the list of **Available CAs**.
6. In the **Available CAs** list, select the local CA name you created and click **Add** to move the CA name to the **Trusted CAs** list.
7. Click **Save**.

Adding a server certificate for the SKM appliance

A server certificate must be created for the SKM appliance.

1. Select the **Security** tab on the SKM key manager.
2. Select **Certificates** under **Certificates and CAs**.

The **Certificate and CA Configuration** page is displayed. This page includes a **Create Request Information** dialog box.

3. Enter the following in the **Create Request Information** dialog box:
 - Certificate Name - HPSKM_Server_029
 - Common Name - HPSKM_Server_029
 - Organization Name - Brocade
 - Organizational Unit Name - Storage Software
 - Locality Name - SJC
 - State or Province Name - CA
 - Country Name - US
 - Email Address - support@brocade.com
 - Key Size - 2048

NOTE

The names shown are examples. You may use other names. Remember the **Certificate Name**, or write it down. You will need it later in the procedure for [“Downloading the local CA certificate file”](#).

4. Select **Create Certificate Request**.

Successful completion is indicated when the new entry for the server certificate appears on the **Certificate List** with a **Certificate Status** of **Request Pending**.
5. Select the pending server certificate from the list.
6. Select **Properties**.

A **Certificate Request Information** dialog box is displayed.
7. Copy the key contents, beginning with `---BEGIN CERTIFICATE REQUEST---` and ending with `---END CERTIFICATE REQUEST---`. Be careful not to include any extra characters.
8. Select **Local CAs** under **Certificates and CAs**.

The **Certificate and CA Configuration** page is displayed.
9. Select the local certificate name from the **CA Name** column.
10. Select **Sign Request**.

A **Sign Certificate Request** dialog box is displayed.
11. Select **Sign with Certificate Authority** using the CA name with a maximum of 3649 days.
12. Select **Certificate Purpose - Server** and enter 3649 as the **Certificate Duration**.
13. Paste the key contents you previously copied in [step 7](#) into the **Certificate Response** window.
14. Select **Sign Request**.

15. Copy the key contents, beginning with `---BEGIN CERTIFICATE REQUEST---` and ending with `---END CERTIFICATE REQUEST---`. Be careful not to include any extra characters.
16. From the **Security** tab, **Certificates and CAs**, select **Certificates**. From the certificate list, select the name of the certificate being signed.
17. Select **Install Certificate**.
18. Paste the certificate data from [step 15](#), and select **Save**. The certificate status is now Active.

Downloading the local CA certificate file

This procedure requires selection of the local certificate authority name (CA name) created using the [“Setting up the local certificate authority”](#) procedure. Have the CA name available so you will be able to select the correct name from the **Local Certificate Authority List**. This procedure also requires you to enter the server certificate name created using the [“Adding a server certificate for the SKM appliance”](#) procedure. Be sure to have the server certificate name available.

1. Select the **Security** tab on the SKM key manager.
2. Select **Local CAs** under **Certificates and CAs**.
The Certificate and CA Configuration page is displayed.
3. Select the local certificate name from the **CA Name** column in the **Local Certificate Authority List**.
4. Select **Download**.
5. After the download completes, save the file locally, and rename the file to change the file extension from `.cert` to `.pem` (e.g., from `hpskm_cal.cert` to `hpskm_cal.pem`).
6. Select the **Device** tab on the SKM key manager.
7. Select **KMS Server** under **Device Configuration**.
The **Key Management Services Configuration** page is displayed.
8. Select **Edit** under **KMS Server Settings**.
9. Click the check boxes for the following:
 - **Use SSL**
 - **Allow Key and Policy Configuration Operations**
 - **Allow Key Export**
10. Type in the server certificate name in the **Server Certificate** field.
11. Select **Save** to save these settings.
12. Select **Edit** under **KMS Server Authentication Settings**.
13. Select **Required** for **Password Authentication**.
14. Select **Save** to save these settings.

Creating an SKM Key vault High Availability cluster

The HP SKM key vault supports clustering of HP SKM appliances for high availability. If two SKM key vaults are configured, they must be clustered. If only a single LKM key vault is configured, it may be clustered for backup purposes, but it will not be directly used by the switch.

To create a cluster, perform the following steps on one of the HP SKM appliances that is to be a member of the cluster

1. Select the **Device** tab on the SKM key manager.
2. Select **Cluster** under **Device Configuration**.
The **Cluster Configuration** page is displayed.
3. Type the cluster password under **Create Cluster**.
The default value for **Local Port** is 9001. This is the recommended value, and should not be changed unless your IT department requires a different value.
4. Select **Create**.
5. Select **Download Cluster Key** under **Cluster Settings**.
6. Copy the cluster key and save it in a convenient location. This key is needed for [“Adding an HP SKM appliance to a cluster”](#). You will be able to browse to the location as part of that procedure.

NOTE

Record the local IP address and cluster password for use in [“Adding an HP SKM appliance to a cluster”](#).

Copying the local CA certificate

1. Select the **Security** tab.
2. Select **Local CAs** under **Certificates & CAs**.
3. Select the name of the local CA from the **Local Certificate Authority** list.
The **CA Certificate Information** is displayed.
4. Copy the key contents, beginning with `---BEGIN CERTIFICATE REQUEST---` and ending with `---END CERTIFICATE REQUEST---`. Be careful not to include any extra characters.
This certificate data will be transferred to other HP SKM appliances in [“Adding an HP SKM appliance to a cluster”](#).
Keep this browser window open while going on to [“Adding an HP SKM appliance to a cluster”](#).

Adding an HP SKM appliance to a cluster

1. Open a new browser window, while keeping the browser window from [“Copying the local CA certificate”](#) open.
2. Log in to the HP SKM Key Manager console of the HP SKM appliance that is being added.
3. Select the **Security** tab.

4. Select **Known CAs** under **Certificates & CAs**.
The **Certificate and CA Configuration** page is displayed.
5. Type the certificate name in the **Certificate Name** field under **Install CA certificate**.
6. Paste the certificate data you copied previously in the “[Copying the local CA certificate](#)” procedure. If you kept the browser window open as suggested in “[Copying the local CA certificate](#)”, the same data is available in that browser window.
7. Select **Install**.
8. From the HP SKM key manager main page, select the **Device** tab.
9. Select **Cluster** under **Device Configuration**.
10. Select **Join Cluster**.
11. Type the original cluster member’s IP address into **Cluster Member IP**. This is the IP address designated as the local IP address that you recorded for this step in “[Creating an SKM Key vault High Availability cluster](#)”
12. Browse to the location of the temporary cluster key file that you copied in “[Creating an SKM Key vault High Availability cluster](#)” for the **Cluster Key File**.
13. Type the cluster password you recorded in “[Creating an SKM Key vault High Availability cluster](#)” as the **Cluster Password**.
14. Select **Join**.
15. You are prompted to confirm the operation. Select **Confirm**.
The **Cluster Configuration** page displays, showing the cluster members.
Repeat the procedure to add more members, as needed. Delete the temporary cluster key file when finished. You should also verify that the same server certificate configured for all cluster members by selecting the **Device** tab, and select **KMS Server Settings**.

Signing the KAC certificate

The KAC certificate exported by the encryption switch or blade must be signed using the certificate authority created in the “[Setting up the local certificate authority](#)” procedure.

1. Go to the location where the `kac_skm_req.csr` file was downloaded on an SCP-capable host. You should have this location recorded and available, as described in “[Exporting the KAC certificate request](#)”.
2. Open the file and copy the contents, beginning with `---BEGIN CERTIFICATE REQUEST---` and ending with `---END CERTIFICATE REQUEST---`. Be careful not to include any extra characters.
3. On the SKM key manager main page, select the **Security** tab.
4. Select **Local CAs** under **Certificates & CAs**.
The **Certificate and CA Configuration** page is displayed.
5. Under **Local Certificate Authority List**, select the CA Name for the CA created in “[Setting up the local certificate authority](#)”.
6. Select **Sign Request**.
The **Sign Certificate Request** page is displayed.

D The HP Secure Key Manager

7. Select **Sign with Certificate Authority** using the CA name with the maximum of 3649 days option.
8. Select **Client** as **Certificate Purpose**.
9. Allow Certificate **Duration** to default to 3649.
10. Paste the file contents that you copied in step 2 in the **Certificate Request Copy** area.
11. Select **Sign Request**.

Upon success, you are presented with the option of downloading the signed certificate.

12. Download the signed certificate to your local system as signed_kac_skm_cert.pem.

This file is then ready to be downloaded to the encryption switch or blade.

Importing a signed certificate (SAN Management program)

The public key certificate from the switch is used to authenticate connections to the key vault.

1. Select a switch from the **Encryption Targets** dialog box, and click the **Properties** tab.

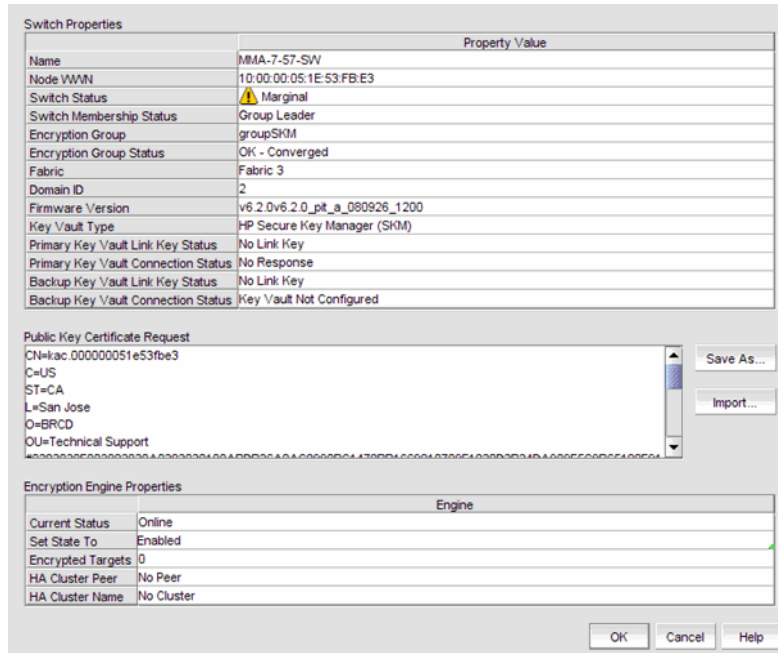


FIGURE 78 Switch Properties dialog box

2. Click the **Import** button.

The **Import Signed Certificate** dialog box displays.

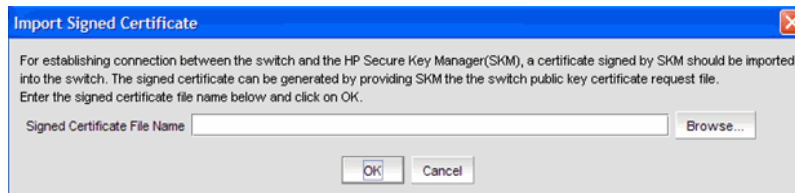


FIGURE 79 Import Signed Certificate dialog box

3. Browse to the location of the stored, signed certificate, and click **OK**.

A connection is now established between the switch and the HP Secure Key Manager (SKM).

SKM key vault high availability deployment

The SKM key vault has high availability clustering capability. SKM appliances can be clustered together in a transparent manner to the end user. Encryption keys saved to one key vault are synchronously hardened to the cluster pairs. Please refer to the HP SKM Appliance user documentation for configuration requirements and procedures.

Configured primary and secondary HPSKM appliances must be registered with the Brocade encryption switch or blade to begin key operations. The user can register only a single SKM if desired. In that case, the HA features are lost, but the archived keys are backed up to any other non-registered cluster members. Beginning with Fabric OS version 6.3.0, the primary and secondary appliances must be clustered.

Both the SKM Appliances in the cluster can be registered using the following command.

```
cryptocfg --reg -keyvault <cert label> <certfile> <hostname/ip address> <primary | secondary>
```

Disk keys and tape pool keys support

DEK creation, retrieval, and update for disk and tape pool keys are as follows:

- **DEK creation** - The DEK is first archived to the virtual IP address of the SKM cluster. The request gets routed to the primary or secondary SKM, and is synchronized with other SKMs in the cluster. If archival is successful, the DEK is read from both the primary or secondary SKMs in the cluster until the DEK is read successfully from both. If successful, then the DEK created can be used for encrypting disk LUNs or tape pool in Brocade native mode. If key archival of the DEK to the SKM cluster fails, an error is logged and the operation is retried. If the failure happens after archival to one of the SKMs, but synchronization to all SKMS in the cluster times out, then an error is logged and the operation is retried. Any DEK archived in this case is not used.
- **DEK retrieval** - The DEK is retrieved from the SKM cluster using the cluster's virtual IP address. If DEK retrieval fails, it is retried.
- **DEK Update** - DEK Update behavior is same as DEK Creation.

Tape LUN support

- **DEK Creation** - The DEK is created and archived to the SKM cluster using the cluster's virtual IP address. The DEK is synchronized with other SKMs in the cluster. Upon successful archival of the DEK to the SKM cluster, the DEK can be used for encryption of the tape LUN. If archival of the DEK to the SKM cluster fails, an error is logged and the operation is retried.
- **DEK retrieval** - The DEK is retrieved from the SKM cluster using the cluster's virtual IP address. If DEK retrieval fails, it is retried.
- **DEK update** - DEK update behavior is same as DEK Creation.

SKM Key Vault Deregistration

Deregistration of either Primary or Secondary LKM KV from an encryption switch or blade is allowed independently.

- **Deregistration of Primary SKM** - You can deregister the Primary SKM from an encryption switch or blade without deregistering the backup or secondary SKM for maintenance or replacement purposes. However, when the primary SKM is deregistered, key creation operations will fail until either primary SKM is reregistered or the secondary SKM is deregistered and reregistered as Primary SKM.

When the Primary SKM is replaced with a different SKM, you must first synchronize the DEKs from the secondary SKM before reregistering the primary SKM.

- **Deregistration of Secondary SKM** - You can deregister the Secondary SKM independently. Future key operations will use only the Primary SKM until the secondary SKM is reregistered on the encryption switch or blade.

When the Secondary SKM is replaced with a different SKM, you must first synchronize the DEKs from Primary SKM before reregistering the secondary SKM.

Thales Encryption Manager for Storage

Communication with the Thales Encryption Manager for Storage (TEMS) is referred to as NCKA in operational descriptions in this appendix. NCKA is secured by wrapping DEKs in a master key. The encryption engine must generate its own master key, send DEKs to NCKA encrypted in the master key, and decrypt DEKs received from NCKA using the same master key. The master key may optionally be stored as a key record in the NCKA key vault as a backup, but NCKA does not assume responsibility for the master key. The master key must be backed up and stored, and policies and procedures for responding to theft or loss must be in place.

The Thales key vault provides a web user interface for management of clients, keys, admins, and configuration parameters. The process for setting up a Brocade encryption switch or blade client consists of the following:

- Creating domains, groups, and clients
- Creating certificates for SSL communication between keyvault and client.

A Thales officer creates domains, groups, and managers (a type of administrator), assigns groups to domains and assigns managers to manage groups. Managers are responsible for creating clients and passwords for the groups they manage.

Generating the Brocade user name and password

The Thales key vaults require that user names and passwords must be configured on every member of an encryption group, using the following command.

```
cryptocfg --reg -KAClogin <primary|secondary>
```

For each node in the encryption group, a different username is generated based on the switch WWN. A password must be configured for this user for the primary and, if configured, the secondary key vault. This user must exist on each configured key vault, and the password for that user must match the password created.

The username and password configuration on the encryption switch or blade should be done before configuring the username and password on the key vault itself. The password on the encryption switch or blade can be changed at any time, as long as the corresponding password is changed on the key vault as well.

Adding a client

Communication must be over an SSL connection. This requires creation of a client certificate signed by a Certificate Authority (CA) on the key vault. It is assumed that a CA has been created by an officer at the keyvault, and a CA certificate has been generated. Also, a group must be created for Brocade by an administrator. This group must exist and is the only supported group for the Brocade encryption switch and blade. Details about how to set up a CA and a group can be found in Thales documentation.

NOTE

Each Thales key vault has both a management IP address and a data IP address. Clients must communicate with the key vaults using the data IP address.

1. Invoke the Thales key vault web browser and log in as manager.
2. Create a group to be used for managing Brocade encryption switches and blades. This group must be named `brocade`. This only needs to be done once for each key vault.
3. Click the **Client** tab.
4. Click the **Add Client** tab.
5. Enter the Brocade user name from the previous procedure “[Generating the Brocade user name and password](#)” in the **Name** field.
6. Enter the password from the previous procedure “[Generating the Brocade user name and password](#)” in the **Password** and **Verify Password** fields.
7. Select the group `brocade` from the **group** menu.
8. Click **Add Client**.

A client user is created. Verify the user just created is listed in the table. Continue with “[Signing the CSR](#)”.

Signing the CSR

1. Export the certificate signing request (CSR) certificate for each encryption group member, using the following command.

```
Cryptocfg -export -scp -KACcsr <host IP> <user name> <file path>
```

NOTE

On some host systems this request does not work. If that is true for your system, copy the .csr file above manually to the workstation you are using to interface with the key vault.

2. Under the **certificate** column in the user table, click on the pen icon for the newly created user.
The **Sign Certificate Request** page is displayed.
3. Either enter the .csr file name exported from the switch in the above steps in the **From file** box, or cut and paste the .csr file contents to the **From text** box and click **sign**.
4. Under the **Certificate** column click on the export icon (globe with an arrow).
A web browser file save dialog displays
5. Click **save** and enter the destination file location for this signed certificate. For example; `brcduser1@ncka-1.pem` for the primary keyvault and `brcduser1@ncka-2.pem` for the secondary keyvault.
6. Perform the above steps for both the primary and secondary key vaults using the same user name, password, and group.

NOTE

the same CSR file is used for both the primary and secondary key vaults; however, the signed certificate exported from the two key vaults are different and must be independently registered as indicated in the steps below.

7. Import the signed certificate back into the switch.

```
cryptocfg -import -scp <local file> <host IP> <host user name> <host file path>
```

NOTE

On some systems the scp (secure copy) may not work, in this case copy the signed certificate file above to: /etc/fabos/certs/mace/

8. Register the signed certificate for each key vault using the following command, specifying either the primary or, if used, the secondary key vault.

```
cryptocfg --reg -KACcert <primary|secondary>
```

9. Repeat steps one through eight for all member nodes in the encryption group.

Registering the certificates

Examples below are for the two Thales key vaults installed. Commands assume the exported signed certificates were saved as `brcduser1@ncka-1` and `brcduser1@ncka-2` for the primary and secondary key vaults and the data port IP addresses are `10.32.44.112` and `10.32.44.114`.

1. Set the key vault type.

```
cryptocfg --set -keyvault NCKA
```

2. Register the signed KAC certificates.

```
cryptocfg --reg -KACcert brcduser1@ncka-1.pem primary
cryptocfg -reg -KACcert brcduser1@ncka-2.pem secondary
```

3. Register the primary and secondary key vault certificates and data port IP addresses.

```
cryptocfg --reg -keyvault NCKA_CA1 brcduser1@ncka-1.pem 10.32.44.112 primary
cryptocfg --reg -keyvault NCKA_CA2 brcduser1@ncka-2.pem 10.32.44.114 secondary
```

NOTE

The signed certificate file contains both the client and keyvault CA certificates so the same file name is used for both the keyvault and KACcert registration.

4. Repeat steps one and two for each encryption group member.
5. Display the group configuration to verify values

```
cryptocfg --show -groupcfg
```

NOTE

The Thales key vault has an active session limit of 32 clients. This includes the Brocade encryption switch and blade, and all other clients. This is not configurable, but must be considered in planning key vault usage.

Thales key vault high availability deployment

Both primary and secondary Thales key vaults must be installed and registered with the Brocade encryption switch or FS8-18 blade before configuring any CryptoTarget containers or LUNs. Installing or registering either primary or secondary Thales NCKA key vault after configuring CryptoTarget containers or LUNs causes DEKs to be out of sync between the primary and secondary key vaults. Thales KM appliances do not support clustering. Dual Thales appliances can be registered with the encryption switch or blade using the following command:

```
cryptocfg --reg -keyvault <cert label> <certfile> <hostname/ip address> <primary | secondary>
```

DEK Creation

DEKs are archived to both the primary and secondary Thales key vaults. Upon successful archival of a DEK onto both primary and secondary KM Appliances, the DEK can be used for encrypting LUNs or Tape-Pools. If archival of a DEK fails for either primary KM Appliance or secondary KM Appliance, an error is logged and DEK creation is retried.

DEK retrieval

The DEK is retrieved from the primary Thales key vault if the primary is online and reachable. If the primary Thales key vault is not online or not reachable, the DEK is retrieved from the secondary Thales key vault.

DEK update

DEK update behavior is same as DEK creation.

Thales key vault deregistration

Deregistration of either Primary or Secondary Thales key vault from the Brocade encryption switch or blade is allowed independently.

Deregistration of the primary Thales key vault - You can deregister the primary Thales key vault from the Brocade encryption switch or blade without deregistering the secondary Thales key vault for maintenance or replacement purposes. However, when the primary Thales key vault is deregistered, key creation operations will fail until either the primary key vault is reregistered or the secondary key vault is deregistered and reregistered as primary.

When the primary key vault is replaced with a different key vault, you must first synchronize the DEKs from the secondary key vault before reregistering the primary key vault.

Deregistration of the secondary Thales key vault - You can deregister the Secondary Thales key vault independently. Future key operations will use only the Primary Thales key vault until the secondary key vault is reregistered back on the Brocade encryption switch or blade.

When the Secondary key vault is replaced with a different key vault, you must first synchronize the DEKs from primary key vault before reregistering the secondary key vault.

Index

A

add commands

- add -haclustermember, 101
- add -initiator, 105, 115, 118
- add -LUN, 111, 118, 126, 129
- add -membervnode, 172

B

Brocade Encryption Switch

See switch

C

certificates

- exporting using the CLI, 93
- exporting, importing, and loading, 13
- file names, 93
- importing using the CLI, 94, 214
- purpose for encryption, 13
- storing the public key, 40
- viewing imported, 95

CLI

- general errors and resolution, 177
- using to configure encryption switch or blade, 81

command RBAC permissions, 83

command validation checks, 82

commands

- ipaddrset, 88
- ipaddrshow, 89
- slotpoweroff, 89
- slotpoweron, 89

commit command, -commit, 170

CommVault Galaxy labeling, 121

configuration

- of encryption group-wide policies, 98
- storage encryption privileges, 17
- tasks for encryption, 2
- warnings about multi-path LUNs, 102, 106, 107, 108, 109, 110, 111, 112, 114

configuring

- Crypto LUNs, 109
- CryptoTarget container, 102
- encrypted storage in a multi-path environment, 59
- HA clusters using the CLI, 100
- key vaults, 99
- smart cards, 18
- tape LUNs using the CLI, 115
- tape pools using the CLI, 120
- tasks to complete before encryption, 81

configuring target ports, 156

connections between a switch and an LKM key vault, 31

container

- adding a LUN to CryptoTarget using the CLI, 109, 110
- creating a CryptoTarget, 105
- deleting a CryptoTarget using the CLI, 107
- discovering a Crypto LUN using the CLI, 109
- moving a CryptoTarget using the CLI, 108
- removing a LUN to CryptoTarget using the CLI, 111
- removing an initiator using the CLI, 107

Control Processor, 82

and RBAC, 82

create commands

- create -container, 105, 115, 117
- create -encgroup, 96
- create -hacluster, 101
- create -tapepool, 123

creating a CryptoTarget container using the CLI, 105

Crypto LUN

- adding to CryptoTarget container using the CLI, 109
- configuring, 109, 110
- modifying parameters, 114
- parameters and policies, 112
- removing, 111

cryptocfg command

- add -haclustermember, 101
- add -initiator, 105, 115, 118
- add -LUN, 111, 118, 126, 129
- add -membernode, 172
- commit, 170
- create -container, 105, 115, 117
- create -encgroup, 96
- create -hacluster, 101
- create -tapepool, 123
- delete -container, 108, 163
- delete -encgroup, 165
- delete -hacluster, 170
- delete -tapepool, 124
- dereg -membernode, 164
- discover -LUN, 118
- discoverLUN, 109, 116
- eject -membernode, 164
- enable -LUN, 115
- enable -rekey, 126
- enable_rekey, 129
- enableEE, 91, 172
- export, 93, 212
- failback -EE, 170
- import, 94, 205, 212, 214
- initEE, 91, 172
- initnode, 91, 172
- leave_encryption_group, 164
- manual_rekey, 127
- modify -LUN, 114, 116, 126, 129
- modify -tapepool, 124
- move -container, 108
- reg -keyvault, 216, 220
- reg -membernode, 96, 172
- regEE, 91, 172
- rem -haclustermember, 163
- rem -LUN, 111
- remove -haclustermember, 166
- remove -initiator, 107
- replace -haclustermember, 167
- replaceEE, 163, 172
- resume_rekey, 128, 187
- set -failback, 98
- set -keyvault, 221
- set -keyvault LKM, 96
- show, 92, 95
- show -container, 106
- show -groupcfg, 207
- show -groupmember, 97, 103, 127, 163
- show -hacluster, 166, 171
- show -tapepool, 123
- zeroize, 90

cryptoCfg commands, permissions for, 83

CryptoTarget container

- adding a LUN, 109, 110
- configuring, 102
- creating, 105
- deleting, 107
- discovering a LUN, 109
- moving, 108
- removing a LUN, 111
- removing an initiator from, 107

D

data re-keying, 125

DEK (data encryption keys), 10, 203

DEK life cycle, 11

delete commands

- delete -container, 108, 163
- delete -encgroup, 165
- delete -hacluster, 170
- delete -tapepool, 124

deployment scenarios

- data mirroring deployment, 143
- deployment as part of an edge fabric, 141
- deployment in fibre channel routed fabrics, 139
- deployment with FCIP extension switches, 142
- dual fabric deployment, 135
- single fabric deployment, 133, 134

deployment with admin domains (AD), 157

deregister command, -dereg -membernode, 164

DF compatibility for disk LUNs, 155

DF compatibility for tapes, 155

DF-compatibility for disk LUNs, 195

DF-compatibility for tape LUNs, 199

DF-compatibility for tape pools, 199

DHCP for IP interfaces, 157

discover commands

- discover -LUN, 118
- discoverLUN, 109, 116

disk metadata, 153

E

eject commands

- eject -membernode, 164

enable a disabled LUN using the CLI, 153

- enable commands
 - enable -LUN, 115
 - enable -rekey, 126
 - enable_rekey, 129
 - enableEE, 172
 - enableEE, 91
- encrypted LUN states, 190
- encryption
 - adding a license, 6
 - adding a target, 35
 - adding new LUNs, 36
 - best practices for licensing, 6
 - configuration planning for the management application, 16
 - configure dialog box, 18
 - configuring
 - LUNs for first-time encryption, 129
 - configuring hosts to access encryption targets, 36
 - configuring in a multi-path environment, 59
 - definition of terms, 3
 - description of blade, 6
 - engines, 5
 - first-time encryption modes, 129
 - frame redirection diagram, 9
 - gathering information before using the setup wizard, 16
 - host and LUN considerations, 1
 - launching the encryption target properties dialog box, 36
 - launching the encryption targets dialog box, 34
 - moving a target to a different encryption engine, 36
 - overview diagram, 8
 - performance licensing for switch, 6
 - physical view of switch, 5
 - removing a target, 35
 - selecting mode for LUNs, 66
 - solution overview, 8
 - switch initialization, 13
 - viewing and editing group properties, 25
- encryption blades
 - port labeling, 88
 - special configuration considerations, 88
- encryption configuration tasks, 2
- encryption engines
 - adding to HA clusters, 29
 - effects of zeroizing, 77
 - recovering from zeroizing, 77
 - removing from HA clusters, 29
 - security processor (SP) states, 189
 - states during an active failover and failback, 170
 - support for tape pools, 32
 - verifying status using the CLI, 92
 - zeroizing, 77
- encryption group
 - adding a member node to using the CLI, 96
 - adding a switch using the management application, 47
 - advanced configuration, 163
 - allowed configuration changes, 176
 - basic configuration, 95
 - configuration impact of split or node isolation, 176
 - confirming configuration status, 44
 - creating using the CLI, 95
 - creating using the encryption setup wizard, 37
 - deleting using the CLI, 165
 - diagnosis DEGRADED status, 93
 - disallowed configuration changes, 176
 - group-wide policy configuration, 98
 - merge and split use cases, 171
 - removing a node using the CLI, 163
 - selecting the key vault type, 39
 - switch connection requirements, 87
 - use cases
 - a member node lost connection to all other nodes in the encryption group, 174
- encryption group properties
 - using the restore master key, 77
 - viewing encryption group properties, 25
- encryption group properties dialog box
 - General tab, 26
 - HA Clusters tab, 29, 51
 - Link Keys tab, 30, 31
 - Members tab, 26
 - Tape Pools tab, 32
- encryption properties
 - viewing properties, 22
- encryption steps
 - adding CryptoTarget containers, 2
 - adding encryption group members, 2
 - configuring LUNs, 2
 - configuring the encryption group leader, 2
 - creating HA clusters, 2
 - creating tape pools, 2
 - initializing the switch, 2
 - registering key vaults with the encryption group leader,

2

setting up and configuring key vaults, 2

encryption switch

definition of, 5

initialization, 90

port labeling, 88

encryption switch or group, removing using the management application, 27

encryption targets

adding to virtual targets and virtual initiators within the encryption switch, 54

configuring hosts for, 61

using the dialog box, 34

using the dialog box to add Disk LUNs, 62

ensure uniform licensing in HA clusters, 157

error recovery instructions

for adding a switch to a new group, 182

for adding a switch to an existing group, 181

error recovery instructions for adding a switch to an

existing group, 181

errors related to the CLI, 177

export commands

-export, 93, 212

F

failback command, --failback -EE, 170

failover and failback, states of encryption engines during, 170

field replaceable unit

See FRU

file names, certificates, 93

firmware download considerations, 148

frame redirection

creating and enabling in an FCR configuration (edge to edge), 141

deploying the encryption switch or blade to hosts and targets, 103

enabling, 103

interop matrix, 201

prerequisites, 103

viewing the zone using the CLI, 106

frame redirection zoning

creating and enabled in a FCR configuration, 140

G

Ge0 and Ge1 ports

assigning IP addresses, 88

group-wide policies, examples using the CLI, 98

H

HA clusters

adding an encryption engine using the CLI, 101

configuration rules, 50, 100

configuring using the CLI, 100

creating, 50

deleting a member using the CLI, 170

displaying configuration using the CLI, 166

limitations, 100

performing a manual failback of an encryption engine using the CLI, 170

removing an encryption engine using the CLI, 166

removing engines from, 51

replacing a member using the CLI, 167

requirements for, 50

swapping engines in, 52

HP SKM, 218

HP-UX considerations, 153

http

[//www.gemalto.com/readers/index.html](http://www.gemalto.com/readers/index.html), 18

I

import commands, --import, 94, 205, 212, 214

initialize commands

--initEE, 172

initEE, 91

--initnode, 91, 172

initializing

encryption switch using the CLI, 90

the switch for encryption, 13

initiators, removing from CryptoTarget container, 107

initiator-target zone, creating, 104

IP addresses

assigning to Ge0 and Ge1ports, 88

K

KEK security processor status, 190

key management system

- LKM, 12
- RKM, 12

key vaults

- adding or changing using the management application, 40
- configuration, 99
- connection from switch, 31
- connections between encryption nodes, 10
- entering the IP address or host name for, 39
- entering the name of the file holding the certificate, 39
- setting up LKM, 96
- setting up RKM, 218

L

labeling

- CommVault Galaxy, 121
- NetBackup, 122
- NetWorker, 122

latency in re-key operations, 159

leave command, --leave_encryption_group, 164

license, adding, 6

licensing

- best practices, 6

Lifetime Key Manager (LKM)

- description of, 204

link keys, creating, 31

LKM

- creating link keys, 31
- key vault setup steps, 96
- support for high availability (HA), 209, 226

LKM key management system, 12

LUN

- adding Crypto LUN to CryptoTarget container, 110
- adding to a CryptoTarget container, 109
- choosing to be added to an encryption target container, 65
- configuration warning, 102, 106, 107, 108, 109, 110,

111, 112, 114, 117

configuring for first-time encryption, 129

configuring for multi-path example, 117

configuring policies using the CLI, 112

editing a re-keying interval, 64

force-enabling for encryption, 115

impact of policy changes, 124

modifying parameters using the CLI, 114

multi-path configuration requirements, 103

policy for DF-compatibility disk LUNs, 195

policy for DF-compatibility tape LUNs, 199

policy for DF-compatibility tape pools, 199

policy parameters, 114

removing Crypto LUN to CryptoTarget container, 111

selecting the encryption mode, 64

setting policy for automatic re-keying, 126

M

manual command, --manual_rekey, 127

manual re-key, 159

master key

active, 67

alternate, 68

backing up, 12

backup, 68

create new master key, 68

creating a new, 76

description of, 67

generating, 12

reasons they are disabled, 68

restore master key, 68

saving to a file, 68

master key usage in RKM environments, 157

modify commands

--modify -LUN, 114, 116, 126, 129

--modify -tapepool, 124

move commands, --move -container, 108

multi-path

configuring Crypto LUN

configuring

for multi-path, 117

LUN configuration example, 117

LUN configuration warning, 114, 117

multi-path configuration for encrypted storage using the Management application, 59

multi-path LUN configuration requirements, 103

multi-path LUN configuration warning, 102, 106, 107, 108, 109, 110, 111, 112

N

NetApp Lifetime Key Manager (LKM), description of, 204
NetApp LKM key vaults
 effects of zeroizing, 77
NetBackup labeling, 122
NetWorker labeling, 122
NS-based transparent frame redirection interop matrix,
 201

P

PID failover, 158
policies
 configuration examples, 98
 for Crypto LUN, 112
 impact of LUN policy changes, 124
 impact of tape pool policy changes, 125
 modifying for LUNs using the CLI, 114
 setting for LUN re-keying, 126
privileges, user, 17

R

redirection zones, 157
register commands
 --reg -keyvault, 216, 220
 --reg -membnode, 96, 172
 --regEE, 172
 regEE, 91
re-keying
 configuring a LUN using the CLI, 126
 definition of offline, 126
 definition of online, 126
 encrypted data on a LUN, 125
 initiating a manual session, 127
 modes, 126
 reasons for suspension or failure, 128, 187
 restrictions, 125
 warning, 127
rekeying policies, 159
remove commands
 --rem -haclustermember, 163
 --rem -LUN, 111
 --remove -haclustermember, 166
 --remove -initiator, 107

replace commands
 --replace -haclustermember, 167
 --replaceEE, 163, 172
restore master key wizard, 77
resume commands
 --resume_rekey, 128, 187
RKM key management system, 12
RKM key vaults
 setting up, 218
role based access control (RBAC) permissions for
 cryptoCfg commands, 83
RSA
 key pair certificates, 93
RSA Key Manager (RKM)
 description of, 212, 218, 232

S

security processor (SP)
 KEK status, 190
 states for encryption engines, 189
security tab on management application
 using to back up a master key, 29
 using to create a master key, 29
 using to restore a master key, 29
set commands
 --set -failback, 98
 --set -keyvault, 221
 --set -keyvault LKM, 96
show commands
 --show, 92, 95
 --show -container, 106
 --show -groupmember, 97, 103, 163
 --show groupmember, 127
 --show -hacluster, 166, 171
 --show -tapepool, 123
show commands --show -groupcfg, 207
SKM, 218
smart cards
 configuring, 18
 removing using the management application, 79
 saving to a file, 79
 tracking using the management application, 79
states
 encrypted LUN, 190

- storage encryption
 - configuration privileges, 17
 - configuring, 55
 - confirming the configuration status, 59
 - selecting the encryption engine for configuration, 56
 - selecting the hosts, 57
 - specifying a name for the target container, 57
- storage encryption security
 - privileges for, 17
- switch encryption configuration
 - confirm configuration using the management application, 48
 - designate switch membership using the management application, 47
 - specify public key certificate filename using the management application, 48
- switch removal, consequences of, 27

T

- tape compression, 154
- tape library media changer considerations, 158
- tape LUN, configuring, 115
- tape metadata, 153
- tape pool
 - impact of policy changes, 125
- tape pools, 154
 - adding, 33
 - CommVault Galaxy labeling using the CLI, 121
 - configuring, 120
 - creating using the CLI, 123
 - deleting using the CLI, 124
 - description of, 32
 - identifying using a name or a number, 33
 - labeling rules, 121
 - modifying, 32
 - modifying using the CLI, 124
 - NetBackup labeling using the CLI, 122
 - NetWorker labeling using the CLI, 122
 - removing, 32
 - tape block zero handling, 154
 - tape key expiry, 155
- terminology for encryption, 3

- troubleshooting
 - cfgshow command, 177
 - configshow, 177
 - cryptocfg --show -groupcfg command, 177
 - cryptocfg --show -groupmember command, 177
 - errdumpall command, 177
 - general encryption using the CLI, 177
 - general errors related to the Configure Switch Encryption wizard, 184
 - management application wizard, 181
 - nsshow command, 177
 - supportsave command, 177
- troubleshooting examples using the CLI, 179
- turn off compression on extension switches, 158
- turn off host-based encryption, 158

U

- user privileges
 - defined, 17
 - resource groups, 17
- using from encryption group properties dialog, 77

V

- validating commands, 82
- verifying encryption engine status using the CLI, 92
- virtual initiators, description of in an encryption configuration, 102
- virtual targets, description of in an encryption configuration, 102

Z

- zeroize command
 - zeroize, 90
- zeroizing
 - effects of using on encryption engine, 77
- zone
 - creating an initiator-target using the CLI, 104

